Mount Rainier, photo by Tomasz Węglarski

CppCon2018: trip report

Michał Kowalczyk

Agenda

- What's CppCon?
- Preparation
- Talks
- Summary

Me

- 6 years as C++ developer
- Senior Software Engineer @ TomTom
- Michi - map visualisation engine
- Interested in cognitive sciences

What's CppCon?

- most famous C++ conference
- 2018@Bellevue, WA
- posters contest, integration party, trainings, lectures, panels, books signing, grill the committee, lightning talks, tool time

Preparation

Preparation

- visa
- transportation
  - book tickets early to reduce costs
  - give yourself enough time to adapt to new environment (jetlag)
- lodgement
  - pool, prefer airb&b over hotels

photo by Tomasz Węglarski

photo by Tomasz Węglarski

My expectations (no context)

- experience something new
    - visit US
    - check the most famous conference in C++ field
- learn something new
- get motivated

My expectations (with context)

- improve my work
  - code better
  - design better
  - debug better
  - look for answers to questions where you cannot easily get expertise (no people with experience in your network)
- generally, get new tools that you can use right after getting back from the conference in your project

Preparation

- 5/7 days of conference, around 200 lectures
- up to 7 competing lectures, got to choose one!
- filter agenda by your areas of interest
- check lectures that have been presented on other conferences (~10%)
- mind maps

Great lectures…

that I skipped

C++ now 2018 MAY 7 - 11 cppnow.org

Attack Vectors

- Arbitrary Code Execution
  - Buffer overruns
  - Code pointer exploits
- Denial Of Service
  - Undefined behavior
- Others
  - SQL Injection
  - Privilege escalation

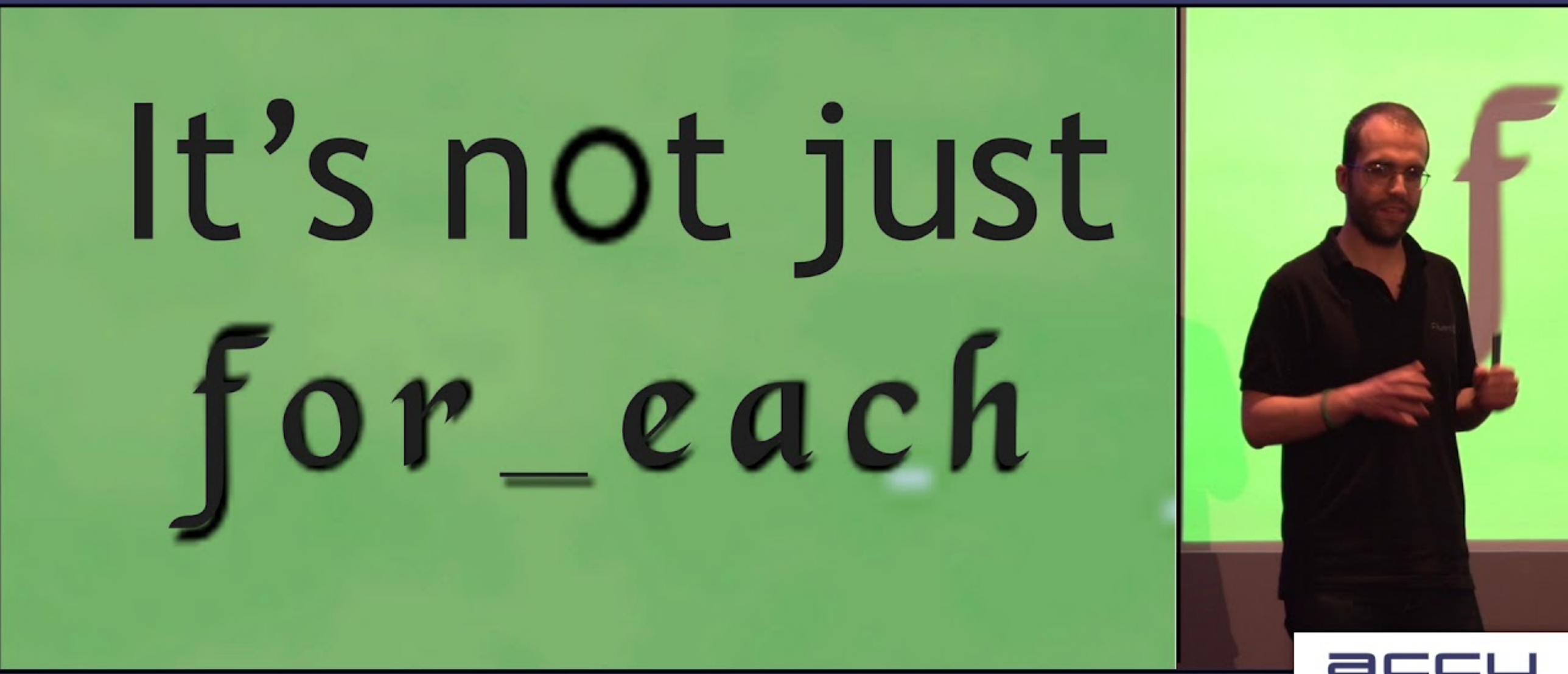
Matthew Butler

Secure Coding Best Practices

www.youtube.com/watch?v=yy8jQgmhbAU
cppcon2018.sched.com/event/FnLW/secure-coding-best-practices-your-first-line-is-the-last-line-of-defense-part-1-of-2

Secure Coding Best Practices: Your First Line Is The Last Line Of Defence by Matthew Butler

- threats, attack surfaces, security layers
- example
- best practises
- warning = error
- fuzz testing
- organisation shift



www.youtube.com/watch?v=bXkWuUe9V2I
cppcon2018.sched.com/event/FnJh/105-stl-algorithms-in-less-than-an-hour

105 STL Algorithms in Less Than an Hour by Jonathan Boccara

- overview through STL algorithms
- great visualisation on a map
- well categorised

C++ now 2018 MAY 7-11 cppnow.org

Ben Deane

Easy to Use,
Hard to Misuse
Declarative Style in C++

Video Sponsorship
Provided By:

JET BRAINS

STATEMENTS

"Except as indicated, statements are executed in sequence." [stmt.stmt] § 1

Properties of statements:

- er…

14 / 122

www.youtube.com/watch?v=2ouxETt75R4
cppcon2018.sched.com/event/FnLS/easy-to-use-hard-to-misuse-declarative-style-in-c

Easy to Use, Hard to Misuse: Declarative Style in C++ by Ben Deane

- expressions vs statements
  - expression
  - type checked
  - verified while compilation
- switch from statements to expressions
  - builder pattern for evaluating validity of an object in compilation time
  - ranges
- critical view on overloading operators in C++ (the standard is wrong in terms of operators priorities)

codeplay
THE HETEROGENEOUS SYSTEMS EXPERTS

How C++ Debuggers Work

Simon Brand
@TartanLlama
Senior Software Engineer, GPGPU Toolchains, Codeplay

Meeting C++ 2017
2017-11-09

www.youtube.com/watch?v=Q3Rm95Mk03c
cppcon2018.sched.com/event/FnJi/how-c-debuggers-work

How C++ Debuggers Work by Simon Brand

- nice overview of binary format on Linux x86_64, including debug symbols (ELF, DWARF)
- overview of low level tools (ptrace)
- understanding of execution of an app on frame level
- how debuggers use of all above and provide typical operations (breakpoints, stepping)

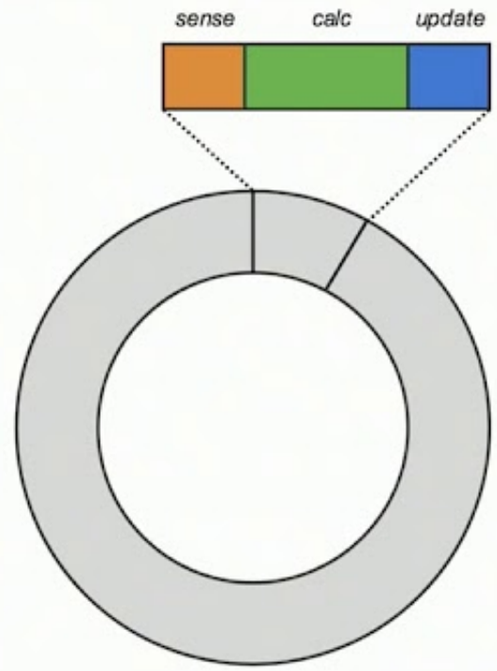
C++ now 2018 MAY 7-11 cppnow.org

Michael Caisse

Modern C++ in an Embedded World

Video Sponsorship Provided By: JET BRAINS

Hard Real-Time



ciere.com

Michael Caisse | Modern C++ in an Embedded World

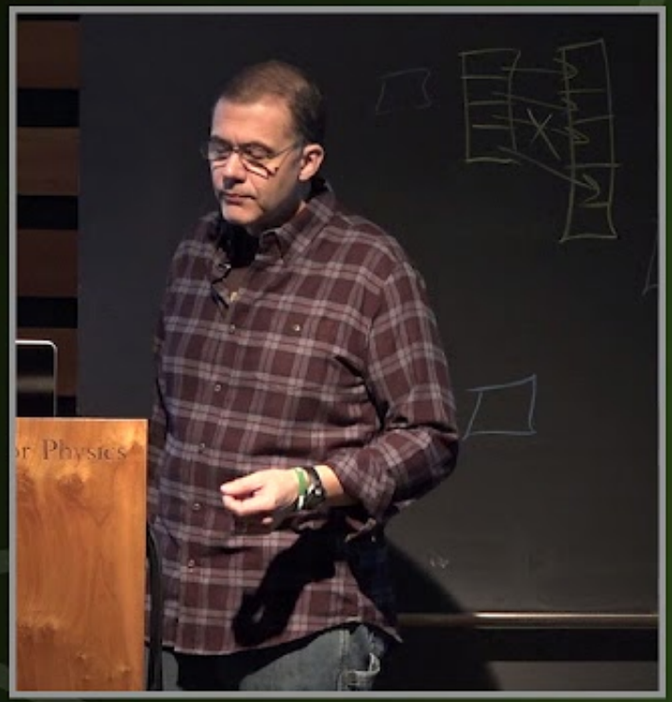
www.youtube.com/watch?v=c9Xt6Me3mJ4

cppcon2018.sched.com/event/FnLk/modern-c-in-embedded-systems-the-saga-continues

Modern C++ in Embedded Systems - The Saga Continues by Michael Caisse

- why C++ in embedded?
  - zero cost abstraction
  - less bugs
  - host debugging
  - take advantage of compiler optimisations
- tooling
  - compiler / linker / IDE
- libraries

C++ now 2018
MAY 7-11
cppnow.org



Bob Steagall

Fancy Pointers for Fun and Profit

Video Sponsorship Provided By:

Possible Traversal-Based Serialization Costs

- In C++, per-type code must be written or generated
  - Traverse source objects and render them to intermediate format
  - Parse the intermediate format and reconstruct destination objects
  - This code can become complex and fragile
- Time – entire stream must be read end-to-end
- Space – many common intermediate formats are verbose
- Private implementation details might be exposed

www.youtube.com/watch?v=Uwe2gXNMeG4
cppcon2018.sched.com/event/FnLH/fancy-pointers-for-fun-and-profit

Fancy Pointers for Fun and Profit by Bob Steagall

- change pointer paradigm so that complex data structures can be simply (de)serialized by memcpy

Great lectures…

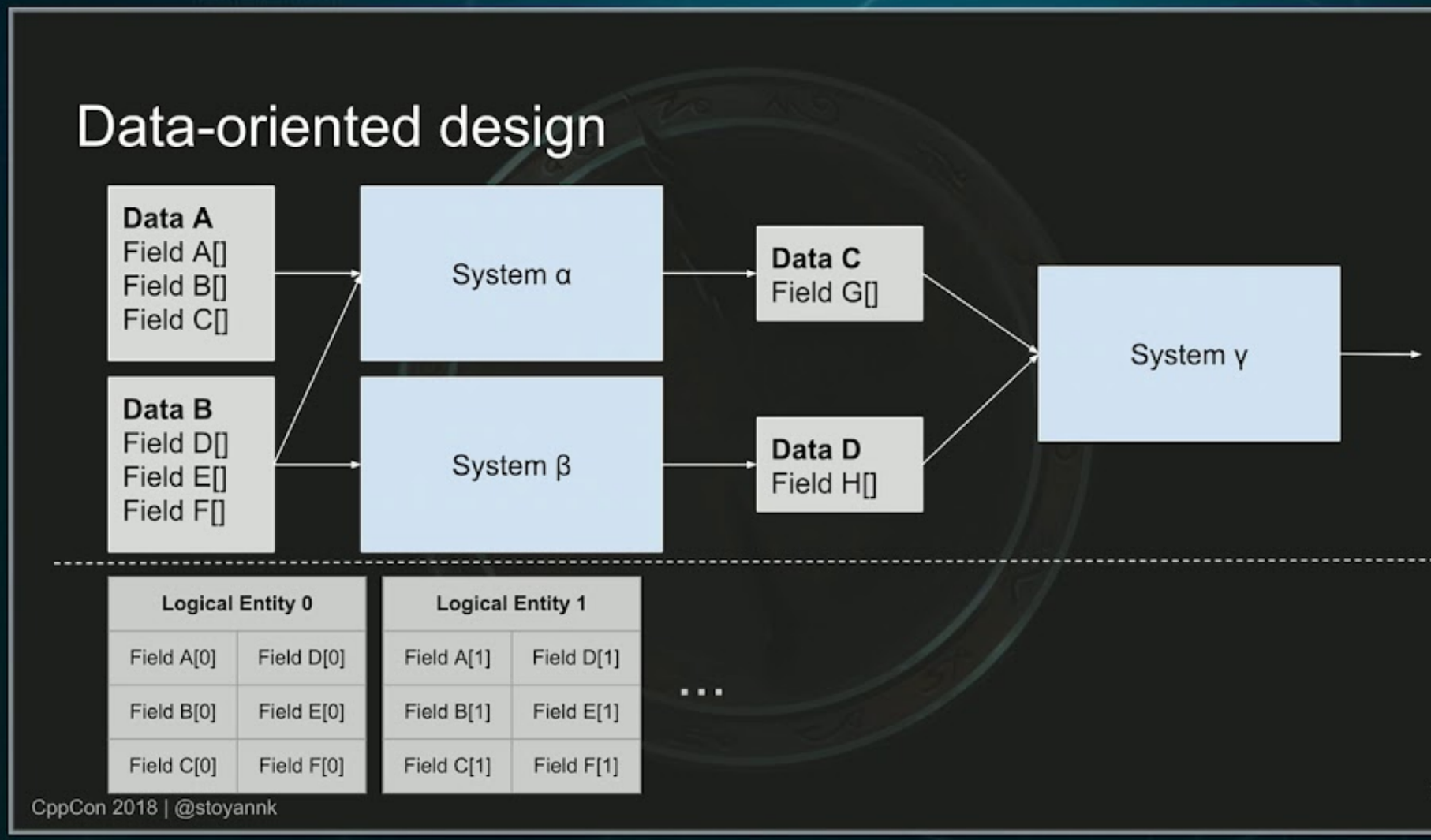
that I attended

Great lectures that I attended

- Programming practices
- Deeper understanding
- Development practices
- Take more advantage of compilers
- Meet others

Programming practices

Data-oriented design



OOP is dead, long live Data-oriented design

CppCon.org

www.youtube.com/watch?v=yy8jQgmhbAU
cppcon2018.sched.com/event/FnKC/oop-is-dead-long-live-data-oriented-design

OOP Is Dead, Long Live Data-oriented Design by Stoyan Nikolov

- example of performance boost on real life system (chromium engine)
- 6 times faster
- reduce cache misses (ifs, virtual calls (on low level))
- create a pipeline (stages of processing instead of intertwining different systems)
- separate processing of different types of data
- drawbacks of dod (quick extensions)

cppcon | 2018

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

JON COHEN
MATT KULUKUNDIS

Touring the Tips of the Week Series

CppCon.org

```
double norm = EvclideanNorm({1, 2.5});
double slope = Slope({point, point});
```

www.youtube.com/watch?v=THDpfWG5T7Y

cppcon2018.sched.com/event/FnLQ/touring-the-c-tip-of-the-week-series

Touring the "C++ Tip of the Week" Series by Jon Cohen & Matt Kulukundis

- abseil.io/tips
  - Tip of the Week #77: Temporaries, Moves, and Copies
  - Tip of the Week #88: Initialization: =, (), and {}
  - Tip of the Week #112: emplace vs. push_back
  - Tip of the Week #142: Multi-parameter Constructors and `explicit`
  - Tip of the Week #122: Test Fixtures, Clarity, and Dataflow

Retrospective

RETROSPECTIVE

Greg Hermanovic

Kim Davidson

ME

1985 1986 1987 1988 1989

SideFX

12

Patterns and Techniques Used in the Houdini 3D Graphics Application

CppCon.org

www.youtube.com/watch?v=2YXwg0n9e7E
cppcon2018.sched.com/event/FnK0/patterns-and-techniques-used-in-the-houdini-3d-graphics-application

Patterns and Techniques Used in the Houdini 3D Graphics Application by Mark Elendt

- C++ in movie industry
- how to manipulate billion points' objects

Pass by value

```
void foo(T);
```

- Snapshot

8

How to
Argue(ment)

www.youtube.com/watch?v=ZbVCGCy3mGQ
cppcon2018.sched.com/event/FnLf/how-to-arguement-what-type-should-i-use-for-my-functions-arguments

How to Argue(ment): What Type Should I Use for My Function's Arguments by Richard Powell

- analysis of all possible ways of passing arguments
- giving meanings:
  - snapshot (by value)
  - view/observer (const l-value ref/ptr)
  - modify (mutable l-value ref/ptr)
  - sink/maybe move (mutable r-value ref&&)
  - optional (pointer T*)

cppcon | 2018
THE C++ CONFERENCE • BELLEVUE, WASHINGTON

Constexpr Edge References Resources Toolchain Impossible
Practices MSVC Learn Nope Code Modules Compiler
Hard to Understand New Features Colleagues
Standard Amount Language Tools Difficulty Evolves
New Stuff Dependencies Older Past Books Difficult to Understand

Simplicity: Not Just For Beginners

CppCon.org

www.youtube.com/watch?v=n0Ak6xtVXno
cppcon2018.sched.com/event/FnKB/simplicity-not-just-for-beginners

Deeper understanding

cppcon | 2018
THE C++ CONFERENCE • BELLEVUE, WASHINGTON

The Bits Between the Bits: How We Get to main()

CppCon.org

SECTIONS

- `.text` — code
- `.rodata` — read-only data
- `.data` — read/write data
- `.bss` — zero-initialised data

www.youtube.com/watch?v=dOfucXtyEsU
cppcon2018.sched.com/event/FnLp/the-bits-between-the-bits-how-we-get-to-main

The Bits Between the Bits: How We Get to main() by Matt Godbolt

- linux specific
- tools: objdump, readelf, gdb, file
- ELF sections (.text, .rodata, .data, .bss)
- object files (.init_array, .text.init)
- static and dynamic linking
- linker script (relocations, link time optimisation)

cppcon | 2018

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

BRYCE ADELSTEIN LELBACH

The C++ Execution Model

CppCon.org

Threads of Execution

NVIDIA

Variables with static storage duration are initialized as a consequence of program initiation. Variables with thread storage duration are initialized as a consequence of thread execution.

[basic.start.static] p1

Main Thread of Execution

Static storage initialization
Thread storage initialization
Evaluate `main()`
Thread storage destruction
Static storage destruction

```
std::thread t(f);
```

Thread storage initialization
Evaluate `f()`
Thread storage destruction

www.youtube.com/watch?v=FJIn1YhPJJc
cppcon2018.sched.com/event/FnLv/the-c-execution-model

The C++ Execution Model
by Bryce Adelstein Lelbach

- excerpts from the standard
  - synchronises with
  - happens before
  - unsequenced
  - indeterminately sequenced

cppcon | 2018
THE C++ CONFERENCE • BELLEVUE, WASHINGTON

Undefined Behavior is Not an Error

CppCon.org

Undefined Behavior

- Example 1
  - consider a recipe which defines how to make a chocolate cake
  - the recipe defines the ingredients required
  - recipe says walnuts are optional and you decide to omit them
    - this is similar to “implementation defined”
  - assume the recipe calls for 1 tsp of salt
    - you add 1 cup of salt
    - the salt is the only issue, however the cake will be awful
    - the entire cake is “undefined behavior”, not just the salt
  - replacing milk with soy milk may work, still undefined behavior

13

www.youtube.com/watch?v=XEXpwis_deQ
cppcon2018.sched.com/event/FnKj/undefined-behavior-is-not-an-error

Undefined Behavior is Not an Error
by Barbara Geller & Ansel Sermersheim

- behavior
  - defined (one meaning)
  - unspecified (many meanings)
  - undefined (no meaning)
- static analysers (tsan, asan, msan, ubsan, coverity, purify)

Dealing with `auto`

- **Couple of ways to initialize an int:**

```
int i1;                  // undefined value
int i2 = 42;             // note: inits with 42
int i3(42);              // inits with 42
int i4 = int();          // inits with 0
int i5{42};              // inits with 42
int i7{};                // inits with 0
int i6 = {42};           // inits with 42
int i8 = {};             // inits with 0

auto i9 = 42;            // inits int with 42
auto i10{42};            // inits int with 42 (unle
auto i11 = {42};         // still inits std::initi
```

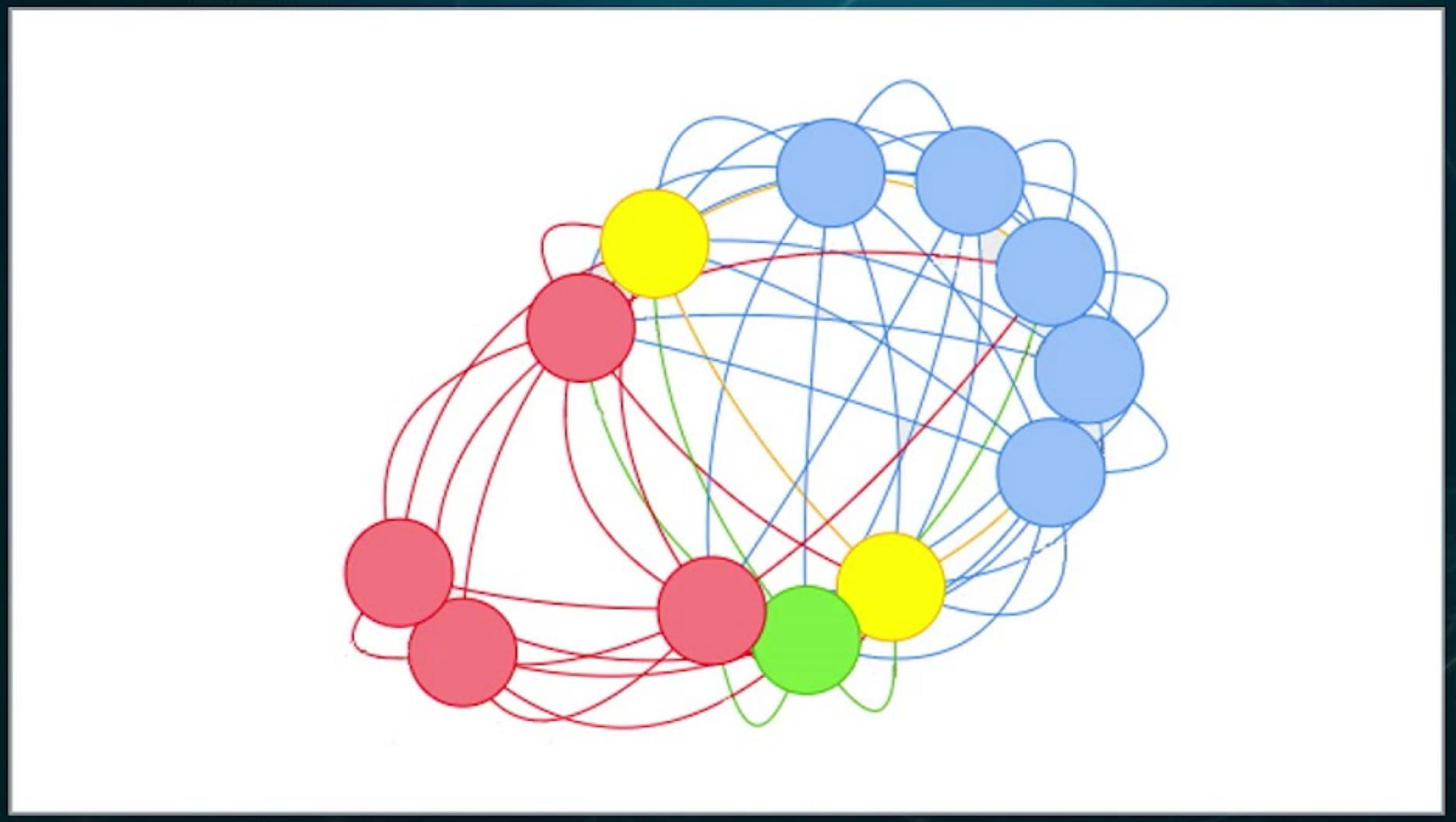
An = in initialization might change the type

The Nightmare of Initialization in C++

www.youtube.com/watch?v=7DTlWPgX6zs
cppcon2018.sched.com/event/FnKp/the-nightmare-of-initialization-in-c

Development practices

cppcon | 2018
THE C++ CONFERENCE • BELLEVUE, WASHINGTON



NICOLAS FLEURY
MATHIEU NAYROLLES

Better C++ using Machine Learning on Large Projects

CppCon.org

www.youtube.com/watch?v=QDvic0QNtOY
cppcon2018.sched.com/event/FnKr/better-c-using-machine-learning-on-large-projects

Better C++ using Machine Learning on Large Projects by Nicolas Fleury & Mathieu Nayrolles

- prevent bugs by deeper understanding of your codebase
- annotate commits
    - introduced bug
    - introduced fix
- use machine learning to predict if future commits are buggy or not
- gamedev industry

cppcon | 2018

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

STEVEN SIMPSON

Source Instrumentation for Monitoring C++ in Production

CppCon.org

printf

"Logging"

16

www.youtube.com/watch?v=0WgC5jnrRx8

cppcon2018.sched.com/event/FnKZ/source-instrumentation-for-monitoring-c-in-production

Source Instrumentation for Monitoring C++ in Production by Steven Simpson

- instrumentation is part of the product
- logging, structured logging
- tracing (start/end, relations, visualisation)
- overhead (turn on/off, sampling)
- metrics (counters, collecting results, combining metrics)
- open tracing (standard)

Development strategies: You've written a library - now what?

CppCon.org

Tests

Disclaimer: I **really** like tests.

A good set of tests can help you over and over again:

1. Make sure that you've fixed a bug
2. Catch regressions
3. Pinpoint problems with someone's installation or configuration.
4. Porting to a new platform

www.youtube.com/watch?v=bfWxNoyRrXs
cppcon2018.sched.com/event/FnK2/development-strategies-youve-written-a-library-now-what

Development strategies: You've written a library - now what? by Marshall Clow

- documentation
- tests
- tools
- compilers
- dealing with users
- managing change

cppcon | 2018
THE C++ CONFERENCE • BELLEVUE, WASHINGTON

The Salami Method for Cross Platform Development

CppCon.org

XAPI: X-PLATFORM PUBLIC C++ INTERFACE

- The *Public* C++ Core API
- Apply Good API Design Principles – ***Focus on the service consumers***
- Consider:
  - Initialization and Shutdown; lifetime management; Sessions; Configuration; Serialization...
- Hides proprietary code/dependencies, reduce dependencies, Pimpl Idiom
- C++ Modules Export Posterchild (C++20)
- Testing: Mock and Unit test the API
- Naming Convention:
  - With core file: `core.cpp` → corresponding: `core_api.cpp`
- May skip for small projects

8

@Adi Shavit :: CppCon 2018 :: videocortex.io

www.youtube.com/watch?v=ZSpGNiUvXVI
cppcon2018.sched.com/event/FnKM/the-salami-method-for-cross-platform-development

The Salami Method for Cross Platform Development by Adi Shavit

- layers
  - core C++
  - C++ public API
  - C wrappers
  - bindings
  - native
  - interface wrapper

cppcon | 2018

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

Undo - Debugging Linux C++

CppCon.org

www.youtube.com/watch?v=V1t6faOKjuQ

cppcon2018.sched.com/event/FnK9/more-gdb-and-other-linux-debugging-wizardry

More gdb and other Linux debugging wizardry by Greg Law

- gdb
  - python support
  - pretty printers
  - dynamic printing
  - remote debugging
- sanitisers (asan, memory sanitiser, thread sanitiser)
- ftrace, strace, ltrace, perf trace
- valgrind

Take more advantage of compilers

C++14/17: WHY?

- Functionality
- Express programmer intent
- Ergonomic metaprogramming
- Convenient syntax sugar

POTENTIAL

- Readability: "Almost like writing python"
- Lower entry barrier for metaprogramming
- More type-safety ⇒ lower maintenance cost

Check out the C++17 Tony tables

Teaching Old Compilers New Tricks: Transpiling C++17 to C++11

www.youtube.com/watch?v=-dtWIY3UMQM
cppcon2018.sched.com/event/FnLc/teaching-old-compilers-new-tricks-transpiling-c17-to-c11

The C++ library

21

Easy::Jit

Just-in-Time
compilation for C++ codes

www.youtube.com/watch?v=_WPdof1dTqo
cppcon2018.sched.com/event/FnKP/easyjit-a-just-in-time-compilation-library-for-c

cppcon | 2018

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

VIKTOR KIRILOV

Interactive C++ Compilation (REPL) Done in a Tiny and Embeddable Way

CppCon.org

Linking to the host application

- for interacting with the host application through the exported API
- the .cpp file always includes a header called "rcrl_for_plugin.h"
  - with helper macros
  - with symbols exported from the host app for persistence

```
RCRL_SYMBOL_IMPORT void*& rcrl_get_persistence(const char* var_name);
RCRL_SYMBOL_IMPORT void   rcrl_add_deleter(void* address, void (*deleter)(void*));
```

- set the ENABLE_EXPORTS CMake property to true on executables

```
set_target_properties(my_executable PROPERTIES ENABLE_EXPORTS ON)
```

- no symbols are exported by default on Windows
  - unlike Unix (with no "-fvisibility=hidden")
  - WINDOWS_EXPORT_ALL_SYMBOLS target property (CMake)

www.youtube.com/watch?v=UEuA0yuw_O0

cppcon2018.sched.com/event/FnJy/interactive-c-compilation-repl-done-in-a-tiny-and-embeddable-way

Meet others

Meet others

- Tool Time, Phil Nash

- C++ Community Building Birds of a Feather, Jon Kalb & Bryce Adelstein Lelbach & Stephan T. Lavavej & Phil Nash & Jens Weller

- Trainers Panel, Michael Caisse & Stephen Dewhurst & Nicolai Josuttis & Scott Meyers

- Run-Time Polymorphism Birds of a Feather, Norman Birkett

- Mock Interviews, Simon Brand

Summary

- a lot of great lectures
- opportunity to meet experts in their fields
- feel being a member of the community

Links

- cppcon2018.sched.com
- youtube.com/user/CppCon
- abseil.io/tips

Thank you

questions?