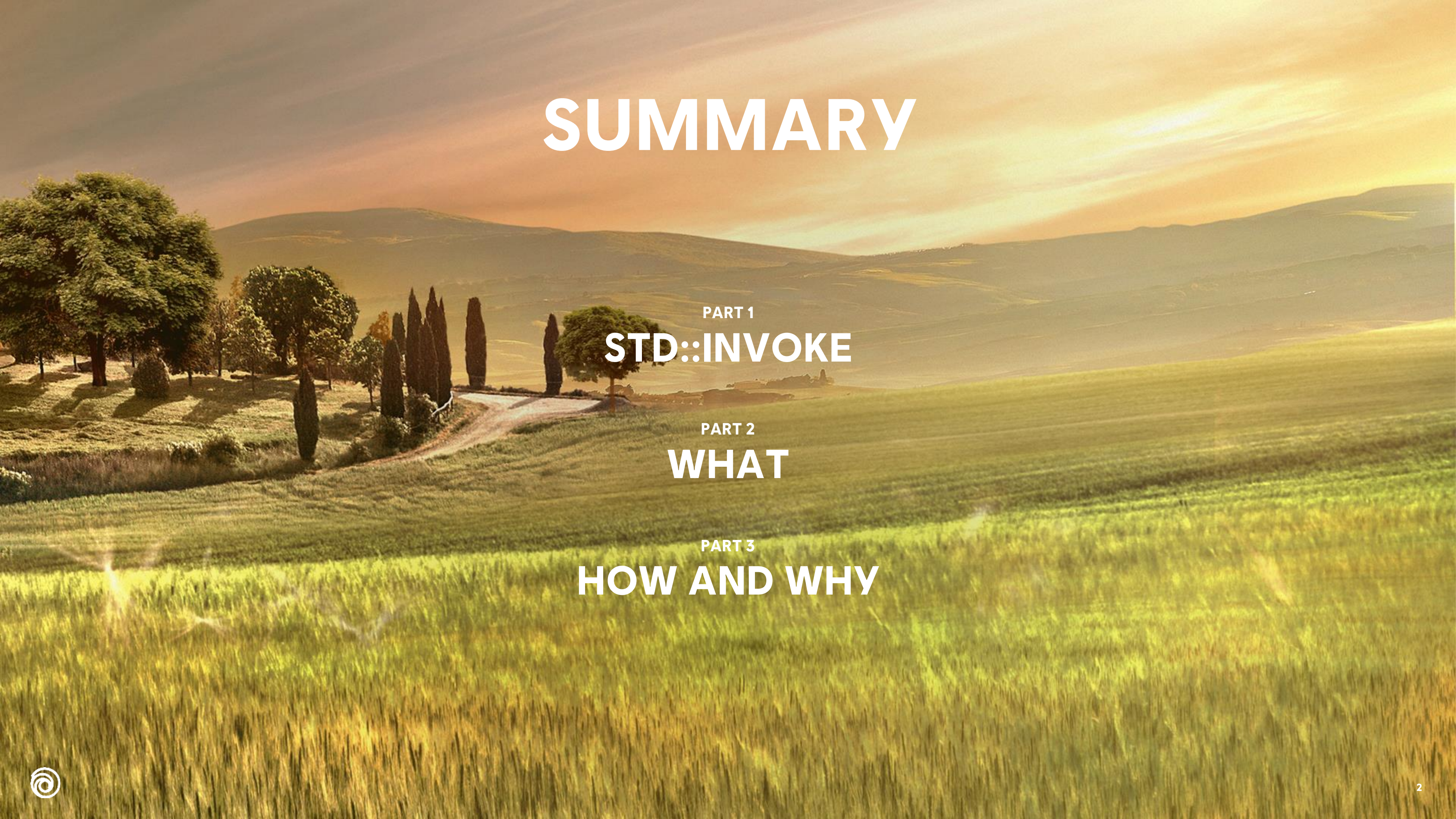
UBISOFT

TEMPLATE METAPROGRAMMING

MADE EASY

MAY 2022



SUMMARY

PART 1

STD::INVOKE

PART 2

WHAT

PART 3

HOW AND WHY

PART 1

STD::INVOKE

std::invoke

So you see, that's where the trouble began.
That function.
That damned function.
imgflip.com

You know, I'm something of a c++ developer myself.

HOW COMPLICATED CAN IT BE?

IT JUST CALLS A FUNCTION

https://en.cppreference.com/w/cpp/utility/functional/invoke

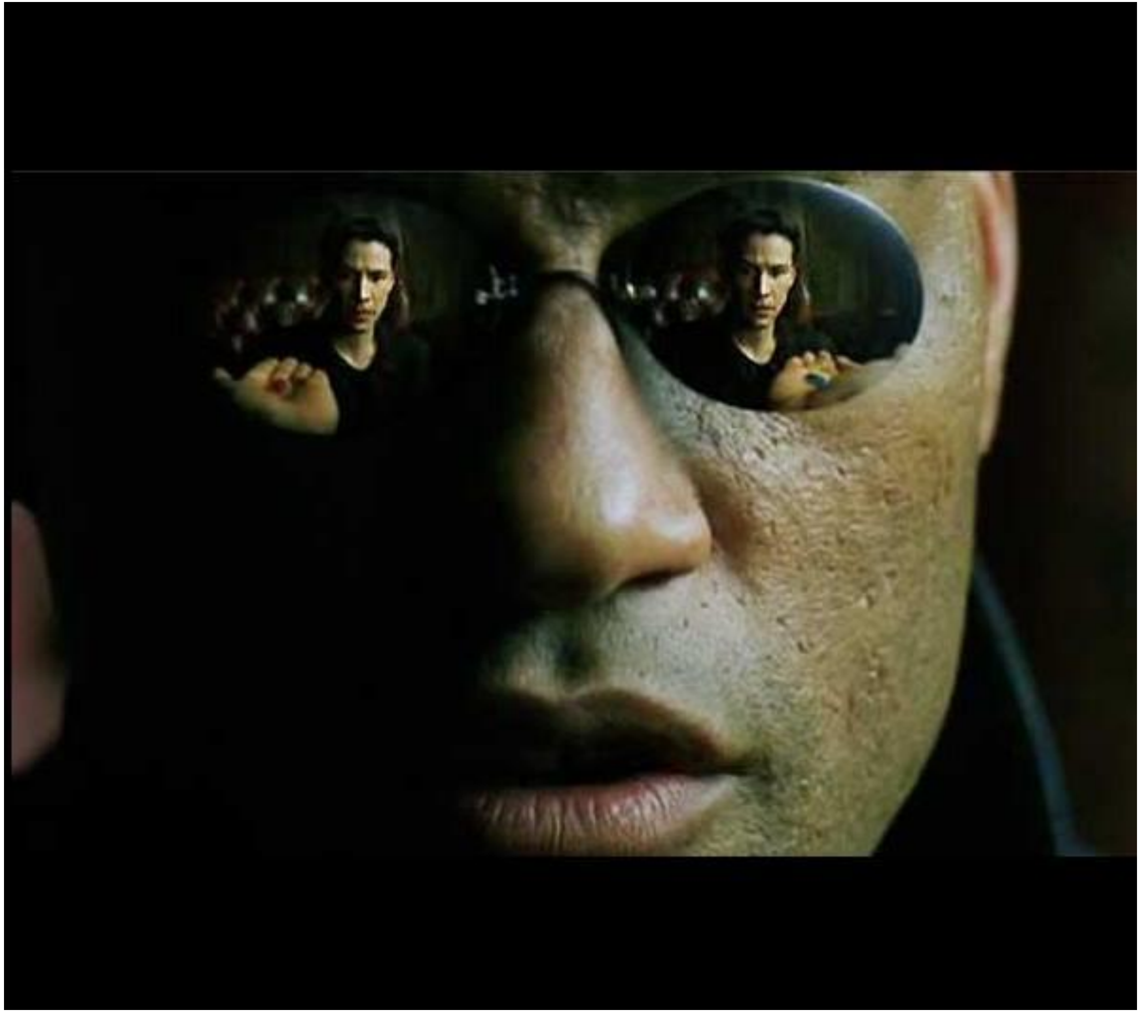
PARAMETERS

f - Callable object to be invoked
args - arguments to pass to f

RETURN VALUE

1) The value returned by f.
2) The value returned by f, implicitly converted to R, if R is not void. None otherwise.

VERY*



PART 2

WHAT

TEMPLATE METAPROGRAMMING

TEMPLATE METAPROGRAMMING

Template metaprogramming (TMP) is a metaprogramming technique in which templates are used by a compiler to generate temporary source code, which is merged by the compiler with the rest of the source code and then compiled.

▼

COMPILE-TIME VERSUS EXECUTION-TIME TRADEOFF

If a great deal of template metaprogramming is used.

GENERIC PROGRAMMING

Template metaprogramming allows the programmer to focus on architecture and delegate to the compiler the generation of any implementation required by client code. Thus, template metaprogramming can accomplish truly generic code, facilitating code minimization and better maintainability

READABILITY

With respect to C++ prior to C++11, the syntax and idioms of template metaprogramming were esoteric compared to conventional C++ programming, and template metaprograms could be very difficult to understand. But from C++11 onward the syntax for value computation metaprogramming becomes more and more akin to "normal" C++, with less and less readability penalty.

INTERMEZZO

FUNCTION TEMPLATES

https://godbolt.org/z/75715xvvG

CLASS TEMPLATES

https://godbolt.org/z/jhYv6MxPn

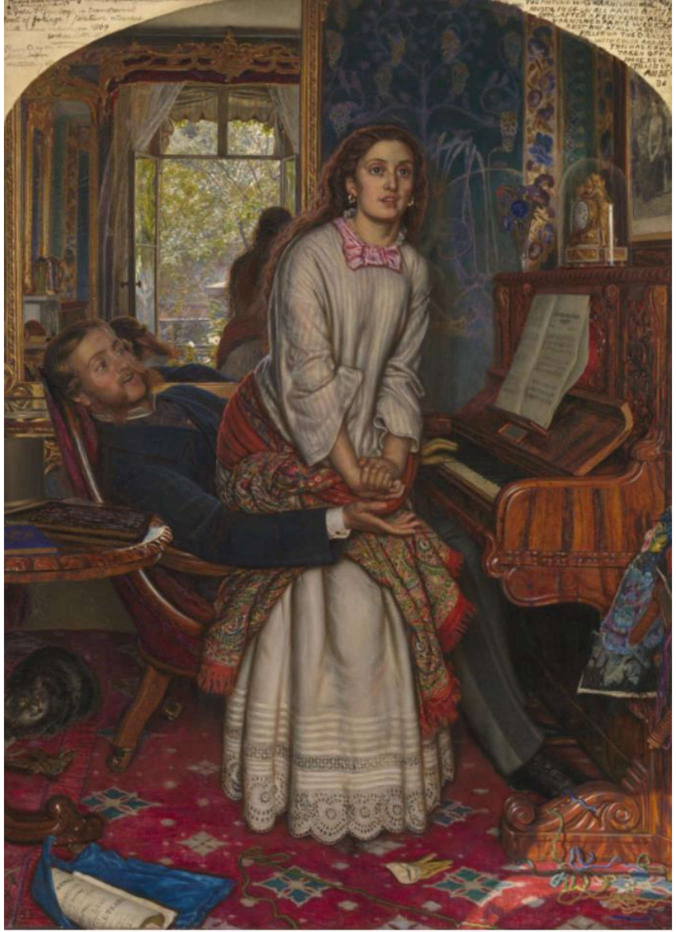
NONTYPE TEMPLATE PARAMETERS

https://godbolt.org/z/jM6EYzMe3

VARIADIC TEMPLATES

https://godbolt.org/z/6ef3dKvKa

HTTPS://GODBOLT.ORG/Z/PVA84VGZV



“Junior developer enthused by consultant’s explanation of the power of C++ templates”

William Holman Hunt

Oil on canvas, 1854

“The dangers of C++ templates: senior dev consoles junior with module that won’t compile after fruitless all-night attempt to deduce the actual problem from the first volume of compiler error messages”

Frank Bramley

Oil on canvas, 1888

PART 3

HOW AND WHY

FUNCTIONAL PROGRAMMING

RECURSION AND EVERYTHING IS CONST

TYPES NOT DATA

LET'S DO IT

EQUALS

FIND VALUE IN AN ARRAY

FIND VALUE IN AN ARRAY AGAIN

FIND VALUE IN AN ARRAY AGAIN AGAIN

https://godbolt.org/z/ah96Go14W

PART 1

STD::INVOKE

BECAUSE IT’S FUN

FURTHER READINGS

C++ TEMPLATES: THE COMPLETE GUIDE-DAVID VANDEVOORDE, NICOLAI JOSUTTIS, DOUGLAS GREGOR

HTTPS://WWW.YOUTUBE.COM/WATCH?V=AM2IS2QCVXY

CPPCON 2014: WALTER E. BROWN "MODERN TEMPLATE METAPROGRAMMING: A COMPENDIUM, PART I"

MODERN C++ DESIGN: GENERIC PROGRAMMING AND DESIGN PATTERNS APPLIED – ANDREI ALEXANDRESCU, DEBBIE LAFFERTY

HTTPS://FATCATCAREERS.COM/SUPPORT-IT-SERBIAN/

Q&A