


Lecture 14:

A Build in the Making

Bart Iver van Blokland

Schedule

Week	Topic
13	Finishing the syllabus
14	Summary lecture
15	Bonus: intro into parallel computing
16	
17	
18	Cool projects



Let me know which topic(s) you want to me to go over in the summary lecture next week!

Inspiraøving

- **Deadline to sign up: this Friday!**
 - 24th and 25th of April at Sluppen
 - Join one of two groups on Blackboard
 - Unable to attend physically?
tdt4102-tilrettelegging@idi.ntnu.no
- Mandatory to take the exam
 - Exception: approved assignments from previous years
- Purpose: become familiar with the exam situation and format.
It's not a test of your skills.

Projects

- Should be delivered on blackboard
- The place to deliver them is only visible after you join a project group
 - Not visible yet, but will appear on the 'Prosjekt' page on Blackboard

Old exams

- Published this week
 - Location: 'Tidl. eksamensoppg.' page on Blackboard

Today

Leftovers!



Today

✓ PROJECT

> .vscode

> builddir

> subprojects

⚙ main.cpp

♥ meson.build

Today

✓ PROJECT

> .vscode

> builddir ← dafuq?

> subprojects ← uw0tm8?

++ main.cpp

♥ meson.build ← WTF?

Today

WHAT THE ACTUAL F



✓ PROJECT

> .vscode

> buildin

> subproj

⚙ main.cp

♥ meson.

GUI Project 0.1

Subprojects

animationwindow : YES

std_lib_facilities: YES 2 warnings

User defined options

buildtype : debug

Today

✓ PROJECT

> .vscode

> builddir

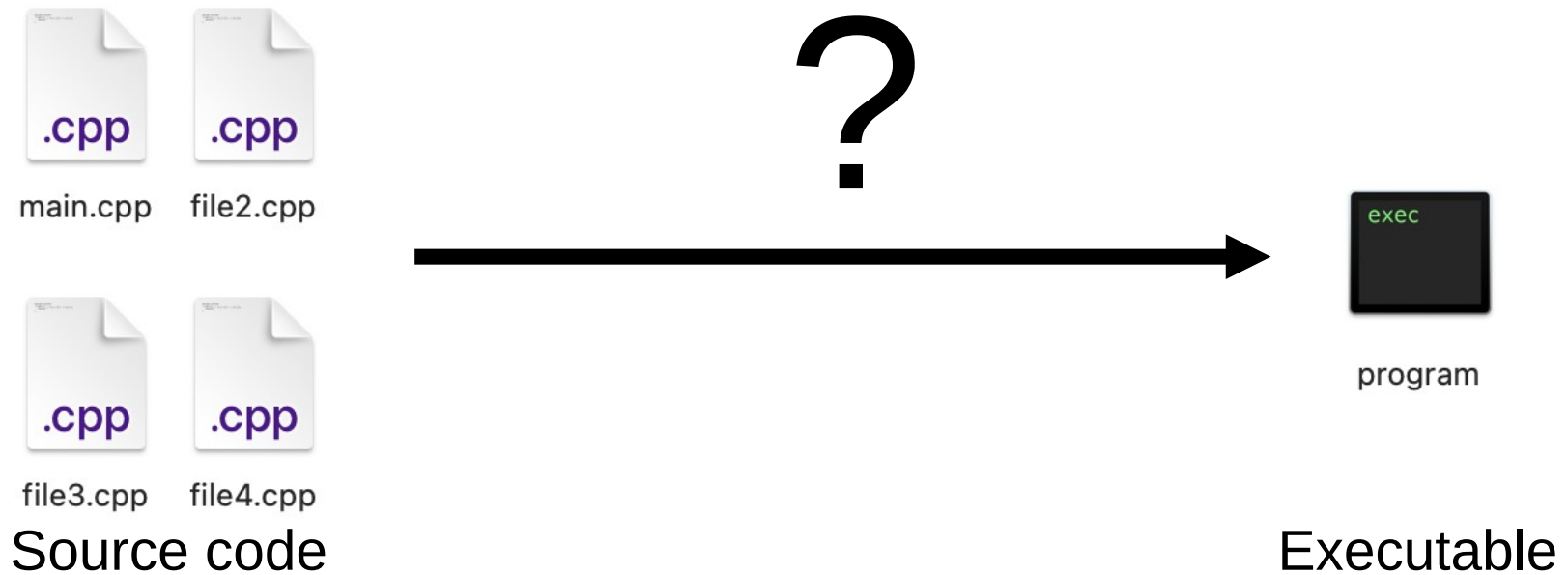
> subprojects

⚙ main.cpp

♥ meson.build

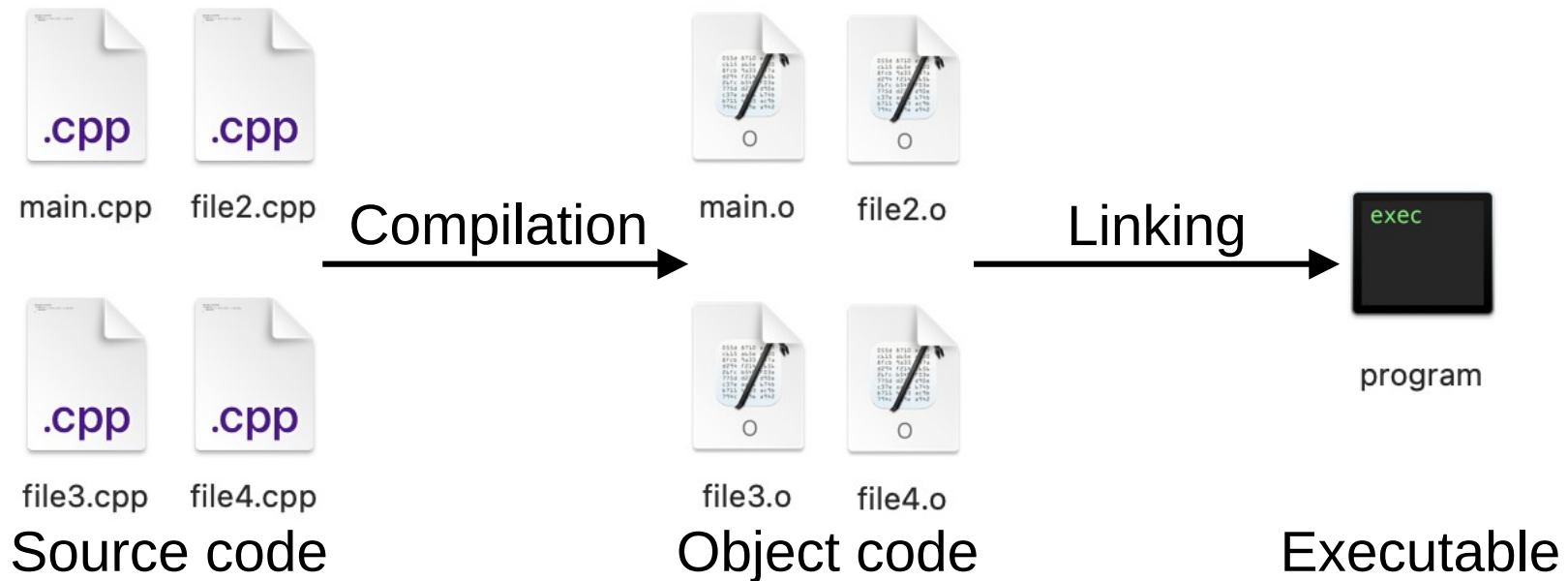
Needed to compile your program.
But why do we need all this?

Reminder: compilation



Reminder: partial compilation

We divide the compilation process into two stages; compilation and linking



Reminder: compilation

- Compile each file separately as much as possible
- Only leaves function calls
- Produces one “object code” file per .cpp file

```
int doWork(int a, int b) {  
    if(isGreater(a, b)) {  
        return add(a, b);  
    } else {  
        return b;  
    }  
}
```

Compilation

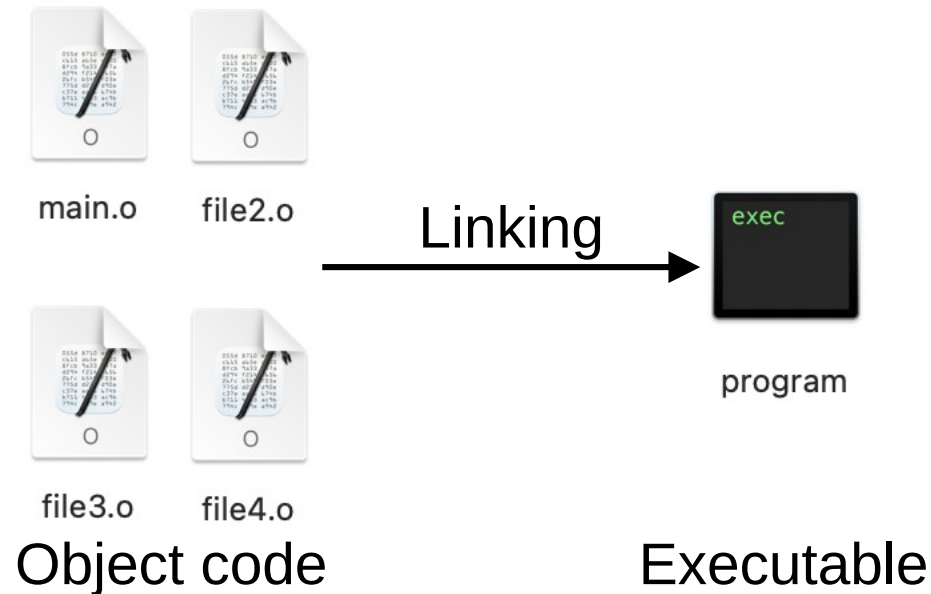
Function calls

```
doWork(int, int):  
    stp    x29, x30, [sp, -32]!  
    mov    x29, sp  
    str    w0, [sp, 28]  
    str    w1, [sp, 24]  
    ldr    w1, [sp, 24]  
    ldr    w0, [sp, 28]  
    bl     isGreater(int, int)  
    and    w0, w0, 255  
    cmp    w0, 0  
    beq    .L6  
    ldr    w1, [sp, 24]  
    ldr    w0, [sp, 28]  
    bl     add(int, int)  
    b      .L7  
.L6:  
    ldr    w0, [sp, 24]  
.L7:  
    ldp    x29, x30, [sp], 32  
    ret
```

<https://godbolt.org/z/1MoTb9Px>

Reminder: linking

- Combine all object code files into an executable file
- Generate the binary instructions needed to do the function calls

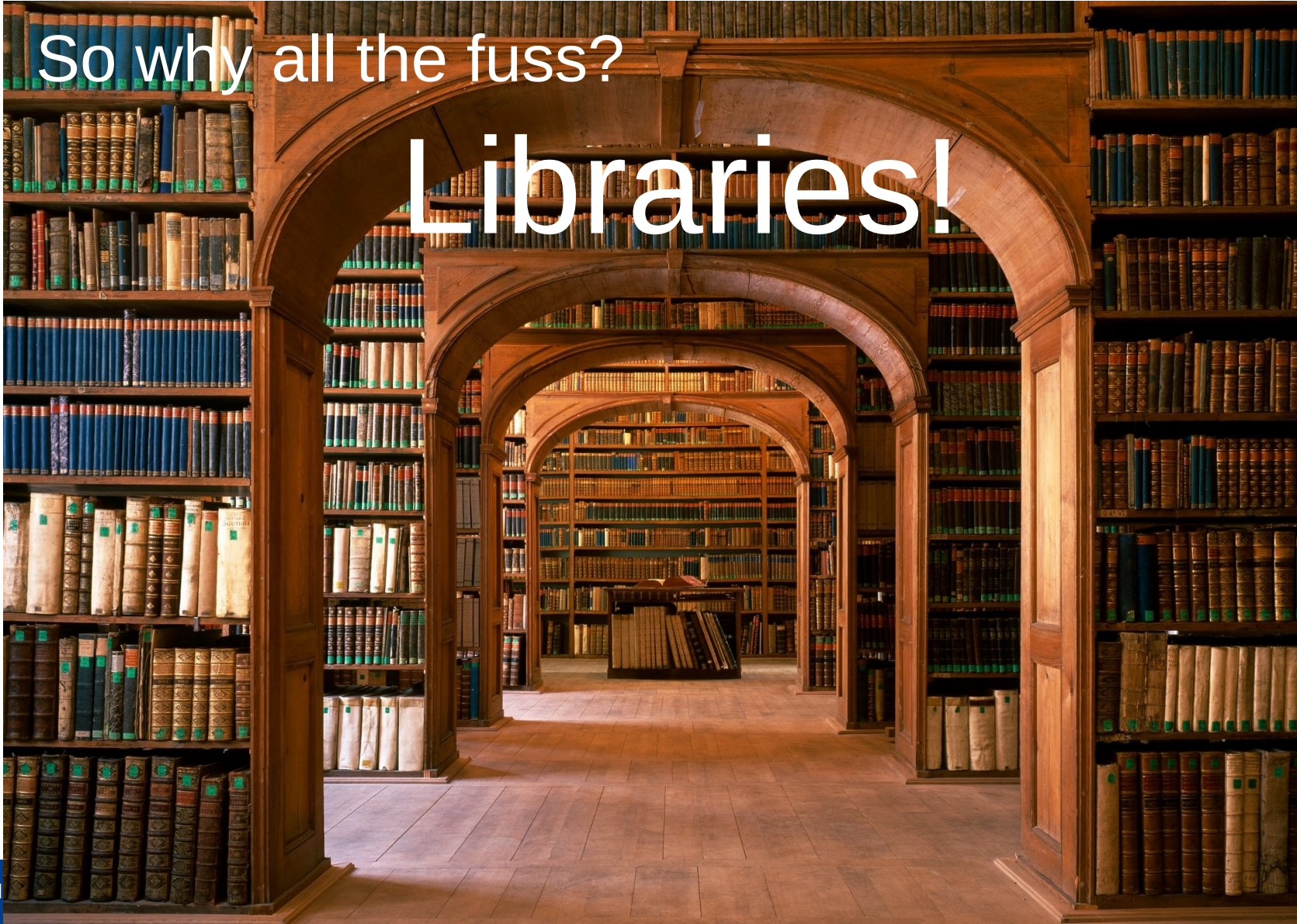


So why all the fuss?

(why do we need Meson?)

So why all the fuss?

Libraries!

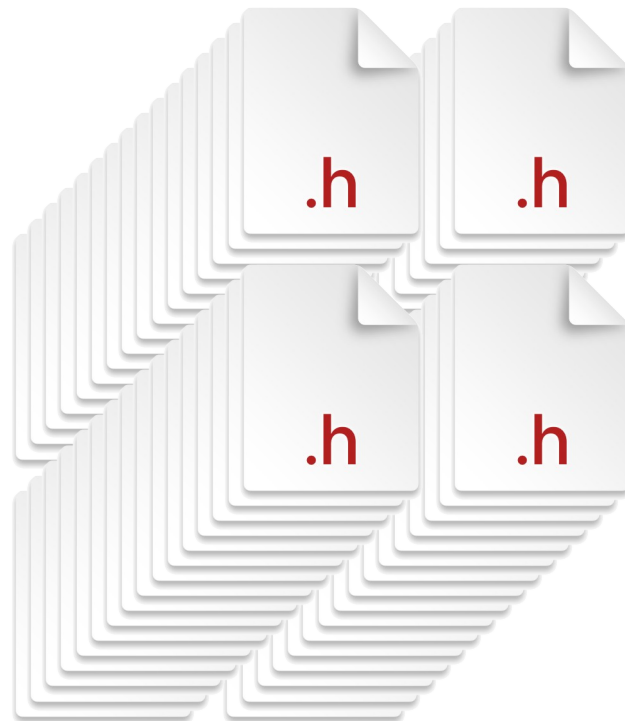


What is a library?



A bunch of C++ files

+



A bunch of header files

What does Meson do?



Your project files

+



Library A



Library B



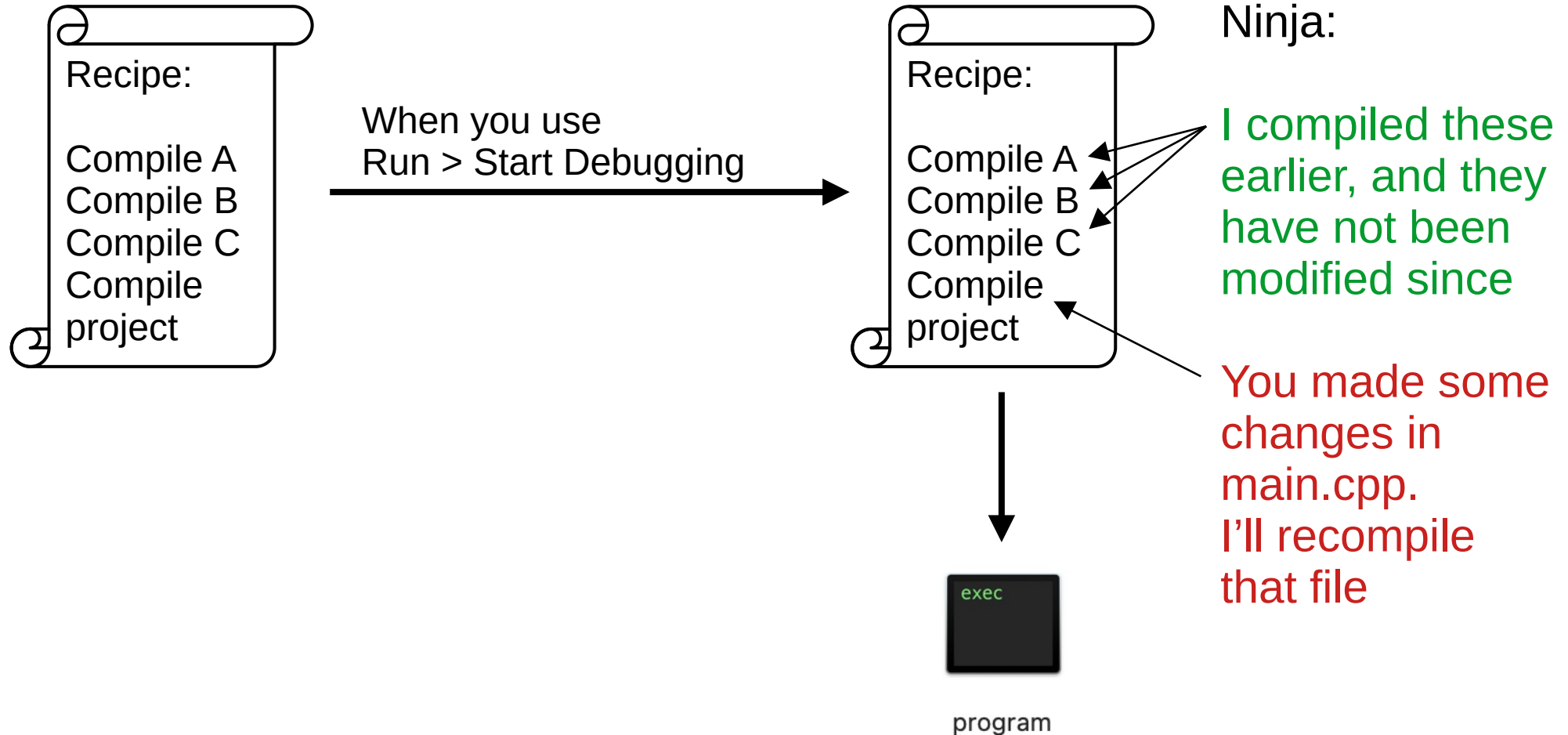
Library C

=

Recipe:

Compile A
Compile B
Compile C
Compile project

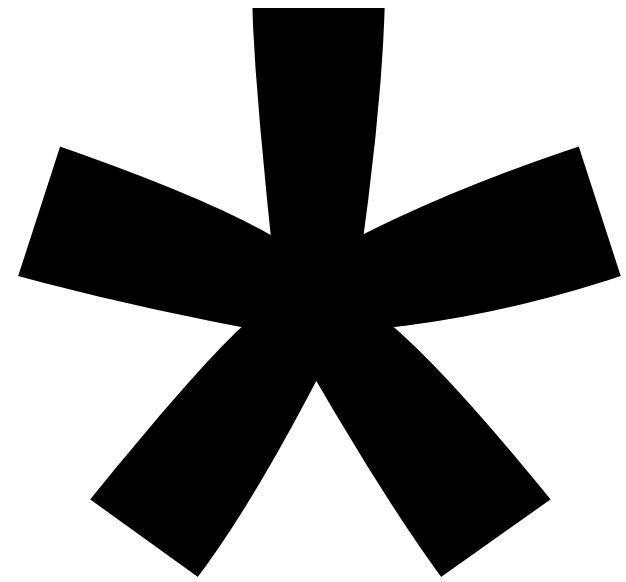
What does Ninja do?



Summary: what does Meson do?

- Create a list of all C++ files that need to be compiled in order to compile the project as a whole
 - Needs doing because of libraries
- A configuration file (meson.build) tells Meson which libraries you want to use.
- Meson produces a 'build recipe' file that can be used by a build tool (we use Ninja) to compile the project

Summary: what does Meson do?



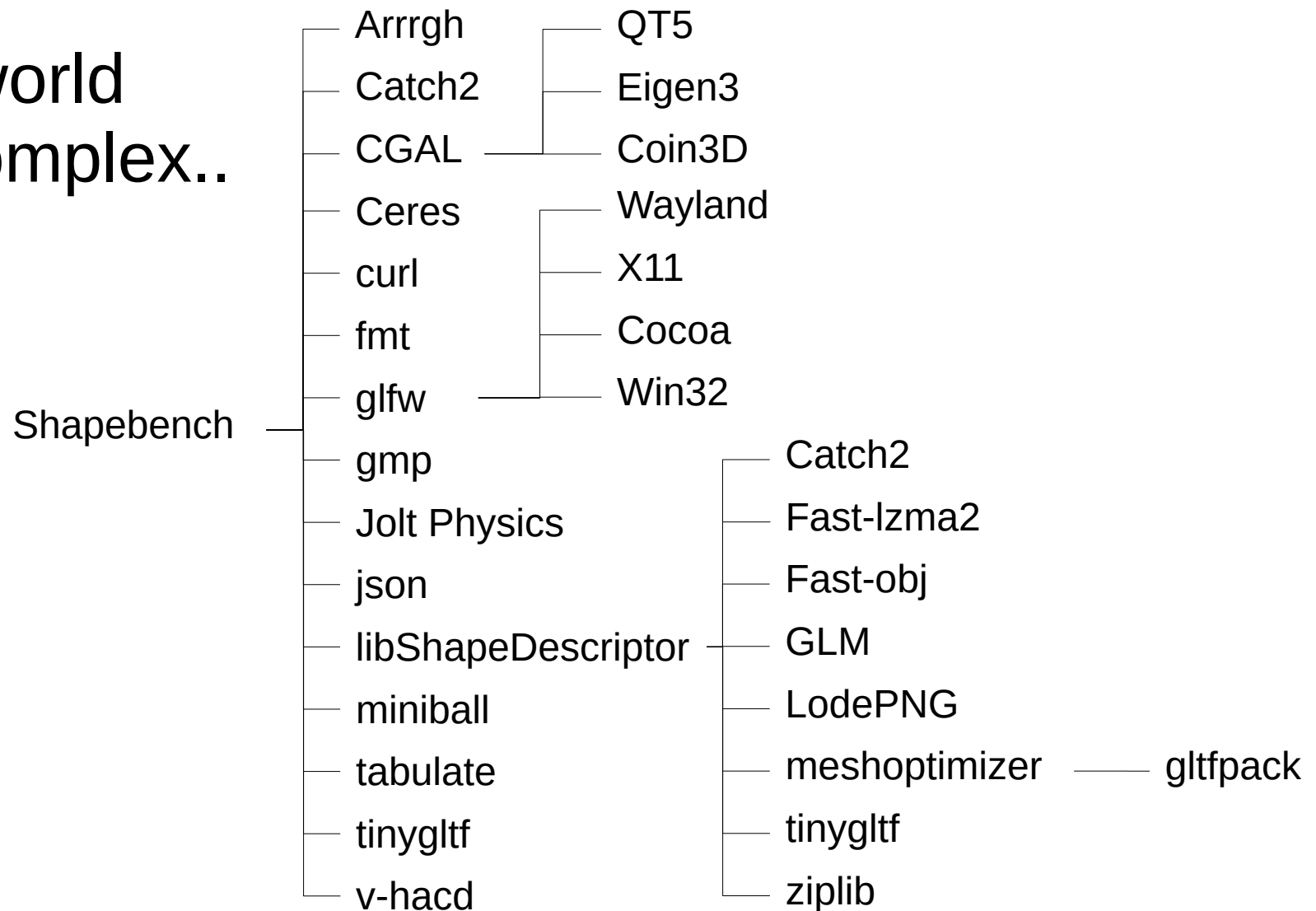
- Create a list of all C++ files that need to be compiled in order to compile the project as a whole
 - Needs doing because of libraries
- A configuration file (meson.build) tells Meson which libraries you want to use.
- Meson produces a 'build recipe' file that can be used by a build tool (we use Ninja) to compile the project

The real world is more complex..

Shapebench

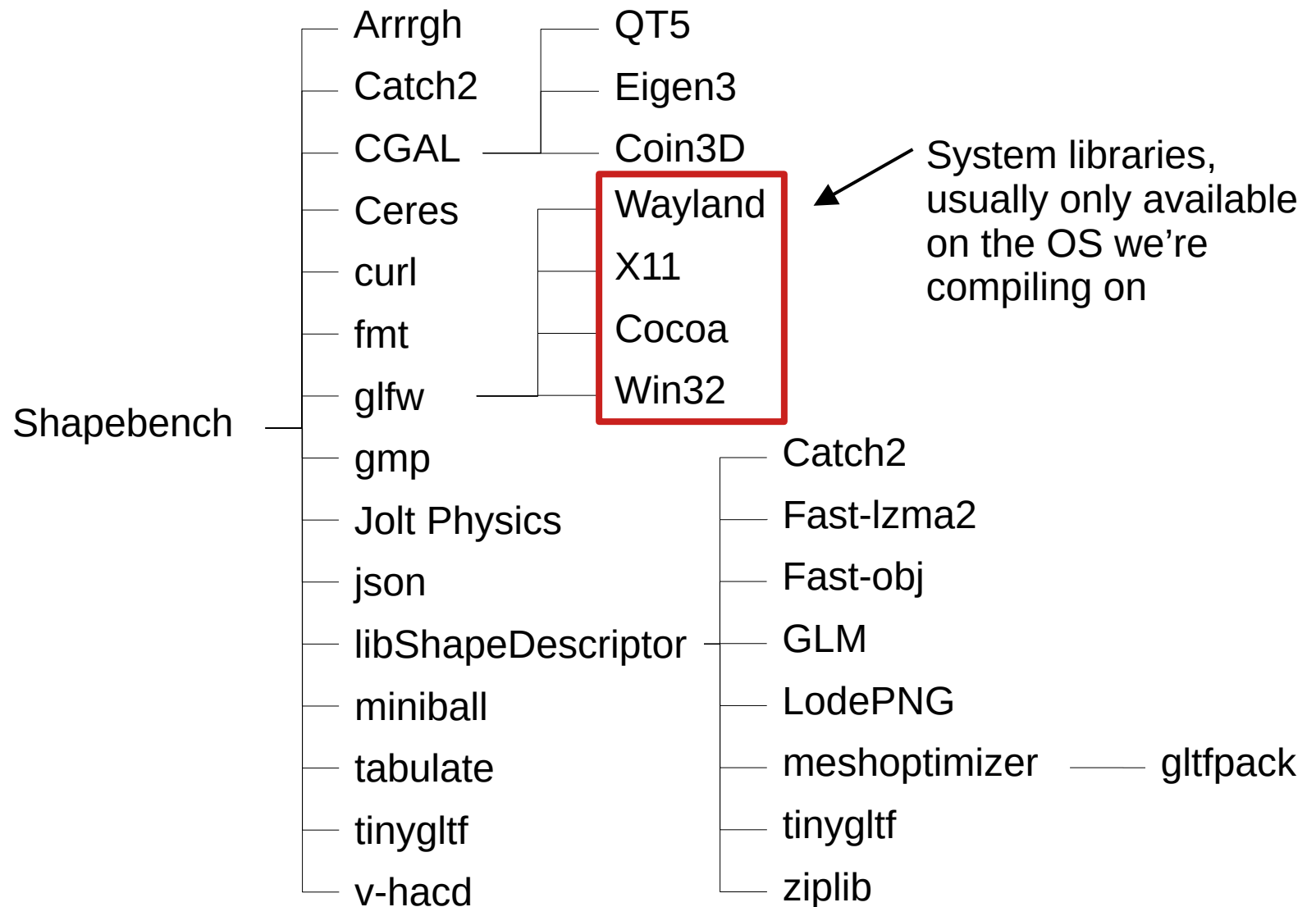
- Arrrgh
- Catch2
- CGAL
- Ceres
- curl
- fmt
- glfw
- gmp
- Jolt Physics
- json
- libShapeDescriptor
- miniball
- tabulate
- tinygltf
- v-hacd

The real world is more complex..



Problems Meson solves:

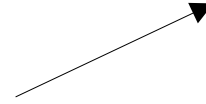
- Manage compilation of libraries, and any libraries used by them



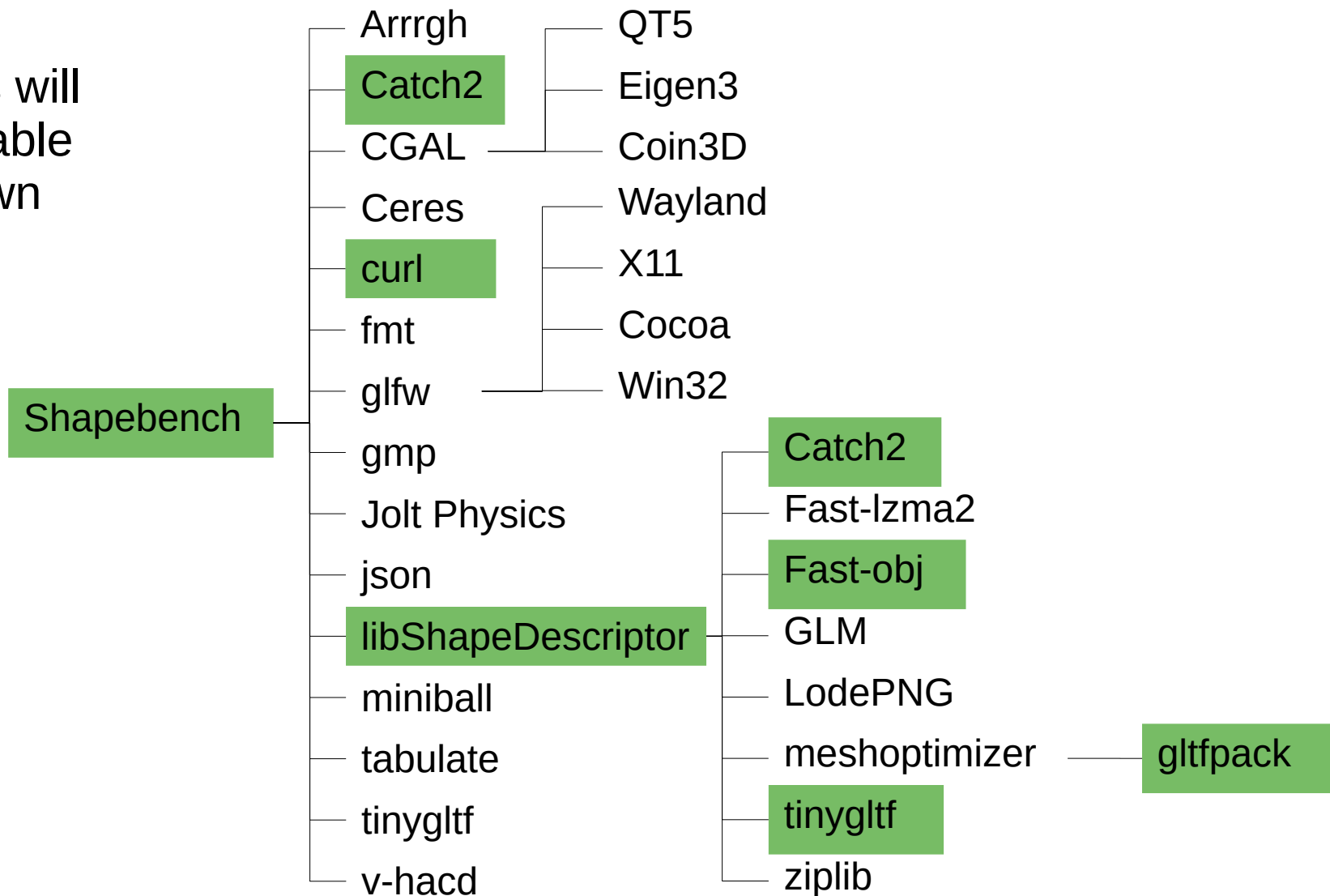
Problems Meson solves:

- Manage compilation of libraries, and any libraries used by them
 - Locate libraries installed on your system through other means

For example package managers!



Some libraries will
create executable
files of their own



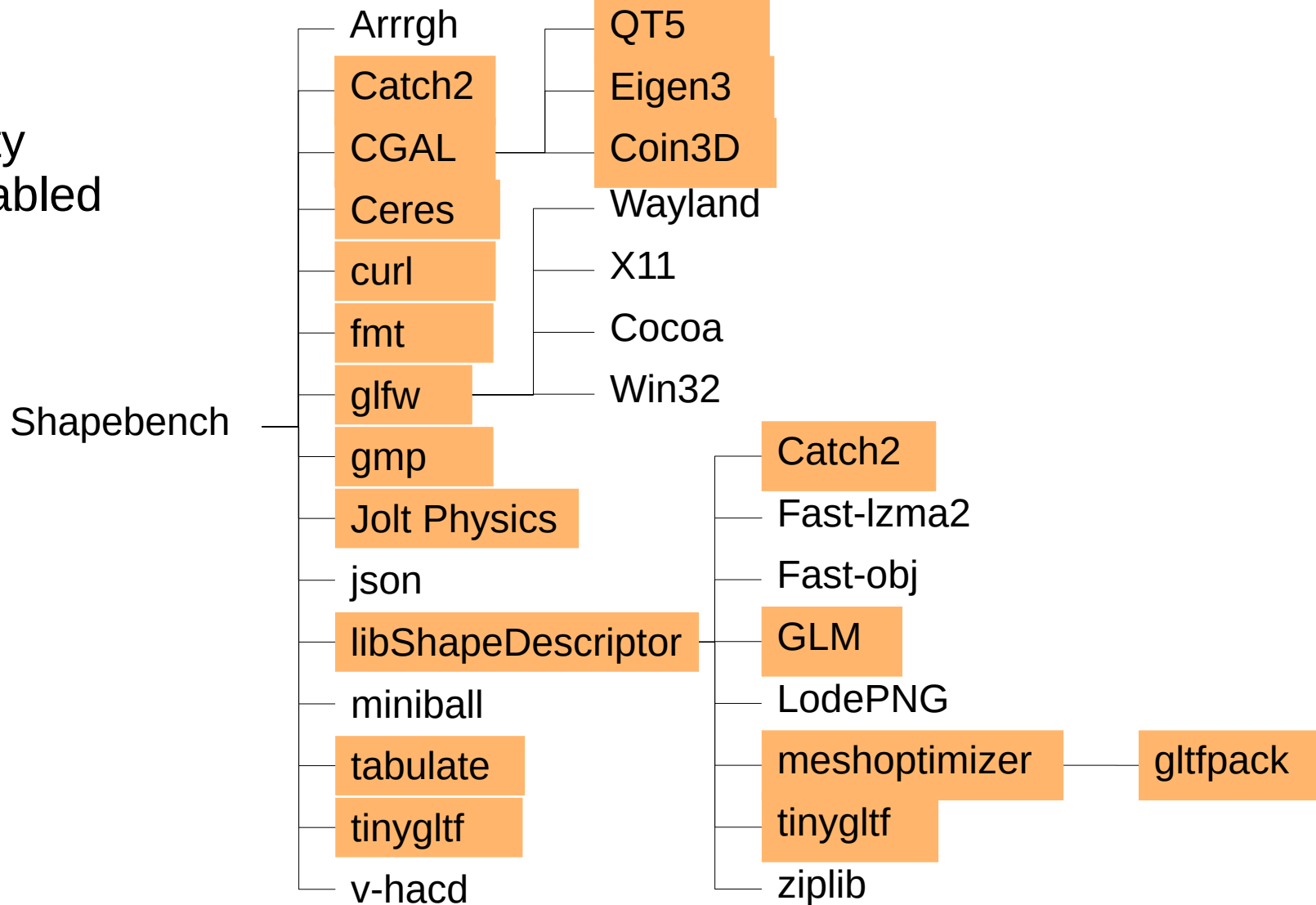
Problems Meson solves:

- Manage compilation of libraries, and any libraries used by them
 - Locate libraries installed on your system through other means
 - Libraries can create executables of their own

Examples:

- Useful utilities that are related to what the library is doing
- Testing frameworks automatically creating executables that let you run your tests
- Examples showing the library in action

Some libraries
have functionality
that can be enabled
or disabled



Problems Meson solves:

- Manage compilation of libraries, and any libraries used by them
 - Locate libraries installed on your system through other means
 - Libraries can create executables of their own
 - Portions of libraries can be enabled or disabled

Examples:

- Tests
- Examples
- Documentation pages
- Support for other specific libraries or tools
- Support for specific hardware features


```
clang++ main.cpp -o program.exe
```

```

/usr/local/cuda/bin/nvcc -forward-unknown-to-host-compiler -DCGAL_DISABLE_ROUNDING_MATH_CHECK
-DCURL_STATICLIB -DDESCRIPTOR_CUDA_KERNELS_ENABLED -DFMT_USE_NONTYPE_TEMPLATE_ARGS=0
-DJPH_CROSS_PLATFORM_DETERMINISTIC -DJPH_DEBUG_RENDERER -DJPH_ENABLE_ASSERTS -DJPH_PROFILE_ENABLED
-DJPH_USE_AVX -DJPH_USE_AVX2 -DJPH_USE_F16C -DJPH_USE_LZCNT -DJPH_USE_SSE4_1 -DJPH_USE_SSE4_2
-DJPH_USE_TZCNT -DNDEBUG -DspinImageWidthPixels=32 -I/home/bart/git/Shapebench-extended/src
-I/home/bart/git/Shapebench-extended/lib/arrgh -I/home/bart/git/Shapebench-extended/lib/tinygltf-2.9.5
-I/home/bart/git/Shapebench-extended/lib/json-3.11.3/include -I/home/bart/git/Shapebench-extended
/lib/miniball/cpp/main -I/home/bart/git/Shapebench-extended/lib/JoltPhysics -I/home/bart/git/Shapebench-
extended/lib/v-hacd/include -I/home/bart/git/Shapebench-extended/lib/glad/include -I/home/bart/git/
Shapebench-extended/lib/fmt-10.2.1/include -I/home/bart/git/Shapebench-extended/lib/tabulate-
master-3a58301067bbc03da89ae5a51b3e05b7da719d38/include -I/home/bart/git/Shapebench-extended/lib/cmake-git
-I/home/bart/git/Shapebench-extended/lib/sha1-1.4 -I/home/bart/git/Shapebench-extended/lib/
libShapeDescriptor/lib/glm-1.0.1 -I/home/bart/git/Shapebench-extended/lib/libcurl-8.7.1/include
-I/home/bart/git/Shapebench-extended/lib/CGAL-5.6.1/include -I/home/bart/git/Shapebench-extended
/lib/PortableGL-0.98.0 -I/home/bart/git/Shapebench-extended/cmake-build-release -I/usr/local/
cuda/targets/x86_64-linux/include -I/home/bart/git/Shapebench-extended/lib/libShapeDescriptor/include
-I/home/bart/git/Shapebench-extended/lib/libShapeDescriptor/lib/cuda-samples/Common -I/home/bart/git
/Shapebench-extended/lib/libShapeDescriptor/lib/meshoptimizer/src -I/home/bart/git/Shapebench-extended
/lib/JoltPhysics/Build/.. -I/home/bart/git/Shapebench-extended/lib/glfw-3.4/include -I/home/bart/git/
Shapebench-extended -I/home/bart/git/Shapebench-extended/lib/ceres-solver-2.2.0/internal/ceres/minilog
-I/home/bart/git/Shapebench-extended/lib/ceres-solver-2.2.0/include -I/home/bart/git/Shapebench-extended
/cmake-build-release/lib/ceres-solver-2.2.0/include -I/home/bart/git/Shapebench-extended/lib/curl-8.7.1
/include -isystem=/usr/include/eigen3 -O3 -DNDEBUG --generate-code=arch=compute_52,code=[compute_52,sm_52]
--generate-code=arch=compute_60,code=[compute_60,sm_60] --generate-code=arch=compute_61,code=[compute_61,
sm_61] --generate-code=arch=compute_70,code=[compute_70,sm_70] --generatecode=arch=compute_75,code=[compute_75
,sm_75] --generate-code=arch=compute_80,code=[compute_80,sm_80] --generate-code=arch=compute_90,code=[compute
_90,sm_90] -Xcompiler=-fopenmp -diag-suppress 20012 -frounding-math -forward-unknown-to-host-compiler
-std=c++20 -DDESCRIPTOR_CUDA_KERNELS_ENABLED -mavx2 -mbmi -mpopcnt -mlzcnt -mf16c -mfpmath=sse -MD -MT
CMakeFiles/shapebench.dir/src/main.cu.o -MF CMakeFiles/shapebench.dir/src/main.cu.o.d -x cu -c
/home/bart/git/Shapebench-extended/src/main.cu -o CMakeFiles/shapebench.dir/src/main.cu.o

```

```

/usr/bin/c++ -O3 -DNDEBUG CMakeFiles/shapebench.dir/lib/glad/src/gl.c.o CMakeFiles/shapebench.dir/src/benchmarkCore
/ComputedConfig.cpp.o CMakeFiles/shapebench.dir/src/benchmarkCore/common-procedures/areaEstimator.cpp.o
CMakeFiles/shapebench.dir/src/dataset/Dataset.cpp.o CMakeFiles/shapebench.dir/src/dataset/DatasetLoader.cpp.o
CMakeFiles/shapebench.dir/src/dataset/LocalDatasetCache.cpp.o CMakeFiles/shapebench.dir/src/dataset/
miniballGenerator.cpp.o CMakeFiles/shapebench.dir/src/filters/FilteredMeshPair.cpp.o CMakeFiles/shapebench.dir/src
/filters/additiveNoise/AdditiveNoiseCache.cpp.o CMakeFiles/shapebench.dir/src/filters/additiveNoise/
OpenGLDebugRenderer.cpp.o CMakeFiles/shapebench.dir/src/filters/additiveNoise/additiveNoiseFilter.cpp.o
CMakeFiles/shapebench.dir/src/filters/gaussianNoise/gaussianNoiseFilter.cpp.o CMakeFiles/shapebench.dir/src/filters
/noisyCapture/NoisyCaptureFilter.cpp.o CMakeFiles/shapebench.dir/src/filters/normalVectorDeviation/normalNoiseFilter
.cpp.o CMakeFiles/shapebench.dir/src/filters/pointCloudResolution/pointCloudResolutionFilter.cpp.o CMakeFiles/
shapebench.dir/src/filters/subtractiveNoise/OcclusionFilter.cpp.o CMakeFiles/shapebench.dir/src/filters/
supportRadiusDeviation/supportRadiusNoise.cpp.o CMakeFiles/shapebench.dir/src/filters/triangleShift
/AlternateTriangulationFilter.cpp.o CMakeFiles/shapebench.dir/src/main.cu.o CMakeFiles/shapebench.dir/src/replication
/RandomSubset.cpp.o CMakeFiles/shapebench.dir/src/results/ResultDumper.cpp.o CMakeFiles/shapebench.dir/src/utils
/AreaCalculator.cpp.o CMakeFiles/shapebench.dir/src/utils/FileCache.cpp.o CMakeFiles/shapebench.dir/src/utils
/FileHasher.cpp.o CMakeFiles/shapebench.dir/src/utils/debugUtils/DebugRenderer.cpp.o CMakeFiles/shapebench.dir/src
/utils/filterUtils/OccludedSceneGenerator.cpp.o CMakeFiles/shapebench.dir/src/utils/filterUtils/cgalConverter.cpp.o
CMakeFiles/shapebench.dir/src/utils/gl/GLUtils.cpp.o CMakeFiles/shapebench.dir/src/utils/gl/GeometryBuffer.cpp.o
CMakeFiles/shapebench.dir/src/utils/gl/Shader.cpp.o CMakeFiles/shapebench.dir/src/utils/gl/ShaderLoader.cpp.o
CMakeFiles/shapebench.dir/src/utils/gl/VAOGenerator.cpp.o CMakeFiles/shapebench.dir/src/utils/methodUtils
/commonSupportVolumeIntersectionTests.cpp.o -o shapebench -L/usr/local/cuda/targets/x86_64-linux/lib/stubs
-L/usr/local/cuda/targets/x86_64-linux/lib -Wl,-rpath,/home/bart/git/Shapebench-extended/lib/gmp-6.3.0/.libs:/usr/
local/cuda/lib64 libShapeDescriptor/libShapeDescriptor.a joltPhysics/libJolt.a lib/glfw-3.4/src/libglfw3.a
libgitinfo.a lib/ceres-solver-2.2.0/lib/libceres.a lib/curl-8.7.1/lib/libcurl.a ../lib/gmp-6.3.0/.libs/libgmp.a
lib/fmt-10.2.1/libfmt.a libShapeDescriptor/fast-lzma2/libfast-lzma2.a libShapeDescriptor/meshoptimizer/
libmeshoptimizer.a /usr/lib/gcc/x86_64-linux-gnu/11/libgomp.so /usr/lib/x86_64-linux-gnu/libpthread.a
/usr/lib/x86_64-linux-gnu/librt.a -lm -ldl /usr/lib/x86_64-linux-gnu/libspqr.so /usr/lib/x86_64-linux-gnu/
libcholmod.so /usr/lib/x86_64-linux-gnu/libamd.so /usr/lib/x86_64-linux-gnu/libcamd.so /usr/lib/x86_64-linux-gnu/
libccolamd.so /usr/lib/x86_64-linux-gnu/libcolamd.so /usr/lib/x86_64-linux-gnu/libsuitesparseconfig.so -lrt
/usr/lib/x86_64-linux-gnu/libtbb.so /usr/local/cuda/lib64/libcudart.so /usr/local/cuda/lib64/libcusolver.so
/usr/local/cuda/lib64/libcublas.so /usr/local/cuda/lib64/libcusparsa.so lib/ceres-solver-2.2.0/lib/
libceres_cuda_kernels.a /usr/lib/x86_64-linux-gnu/liblapack.so /usr/lib/x86_64-linux-gnu/libblas.so /usr/lib/x86_64-
linux-gnu/libssl.so /usr/lib/x86_64-linux-gnu/libcrypto.so /usr/lib/x86_64-linux-gnu/libz.so
-lcudadevrt -lcudart_static -lrt -lpthread -ldl && :

```

```

##### -amdpgpu-arch-tool -analyzer-output -analyze -cmt-migrate-emmit-errors -arcmt-migrate-report-output -B -b -cc1-args -CC -cl-denorms-are-zero -cl-ext -cl-fast-relaxed-math -cl-finite-math-only -cl-fp32-
correctly-rounded-divide-sqrt -cl-kernel-arg-info -cl-mad-enable -cl-no-signed-zeros -cl-no-stdc -cl-opt-disable -cl-single-precision-constant -cl-std -cl-strict-aliasing -cl-uniform-work-group-size -cl-unsafe-math-
optimizations --config -cuda-compile-host-device --cuda-device-only --cuda-feature --cuda-host-only --cuda-include-ptx --cuda-nvopt-device-debug --cuda-path-ignore-env --cuda-path-cuid -cxx-isystem -C -c -darwin-
target-variant-triple -darwin-target-variant -dD -dependency-dot -dependency-file -dI -dM -dsym-dir -dump-depscan-tree -dumppdir -D -emit-ast --emit-extension-symbol-graphs -emit-interface-stubs -emit-llvm -emit-
merged-ifs --emit-static-lib -emit-symbol-graph --end-no-unused-arguments --extract-api -E -faapcs-bitfield-load -faapcs-bitfield-width -faddrsig -falign-loops -faligned-allocation -follow-editor-
placeholders -faltivec-src-compat -fansi-escape-codes -fapinotes-cache-path -fapinotes-modules -fapinotes-swift-version -fapinotes -fapple-kext -fapple-link-rtlib -fapple-pragma-pack -fapplication-extension -fapprox-
func -fasync-exceptions -fbasic-block-sections -fbinutils-version -fblocks -fborland-extensions -fbranch-target-identification -fbuild-session-file -fbuild-session-timestamp -fbuiltin-headers-in-system-modules
-fbuiltin-module-map -fc++-abi -fcall-saved-x10 -fcall-saved-x11 -fcall-saved-x12 -fcall-saved-x13 -fcall-saved-x14 -fcall-saved-x15 -fcall-saved-x18 -fcall-saved-x8 -fcall-saved-x9 -fcaret-diagnostics-max-lines -fcf-
protection -fchar8_t -fcheck-new -fclang-abi-compat -fcolor-diagnostics -fcomment-block-commands -fcommon -fcomplete-member-pointers -fconstexpr-backtrace-limit -fconstexpr-depth -fconstexpr-steps -fcoro-aligned-
allocation -fcoroutines -fcoverage-compilation-dir -fcoverage-mapping -fcoverage-prefix-map -fcrash-diagnostics-dir -fcrash-diagnostics -fcs-profile-generate -fcs-profile-generate -fcuda-approx-
transcendentals -fcuda-short-pttr -fcxx-exceptions -fcxx-modules -fdatasections -fdebug-compilation-dir -fdebug-default-version -fdebug-info-for-profiling -fdebug-macro -fdebug-prefix-map -fdebug-ranges-base-address
-fdebug-types-section -fdeclspec -fdefine-target-os-macros -fdelayed-template-parsing -fdelete-null-pointer-checks -fdepscan-daemon -fdepscan-include-tree -fdepscan-prefix-map-sdk -fdepscan-prefix-map-toolchain
-fdepscan-prefix-map -fdepscan-share-identifier -fdepscan-share-parent -fdepscan-share-parent -fdepscan-share-stop -fdepscan-share -fdepscan -fdiagnostics-absolute-paths -fdiagnostics-hotness-threshold
fdiagnostics-misexpect-tolerance -fdiagnostics-parseable-fixits -fdiagnostics-print-source-range-info -fdiagnostics-show-hotness -fdiagnostics-show-note-include-stack -fdiagnostics-show-option -fdiagnostics-show-
template-tree -fdigraphs -fdirect-access-external-data -fdiscard-value-names -fdollars-in-identifiers -fdriver-only -fdwarf-exceptions -felminate-unused-debug-types -fembed-bitcode-marker -fembed-bitcode -fembed-
bitcode -fembed-offload-object -femit-all-decls -femit-compact-unwind-non-canonical -femit-dwarf-unwind -femulated-tls -fenable-matrix -fexceptions -fexcess-precision -fexperimental-library -fexperimental-new-constant-
interpreter -fexperimental-relative-c++-abi-vtables -fexperimental-sanitize-metadata -fexperimental-sanitize-metadata -fexperimental-strict-floating-point -fextend-arguments -ffast-math -ffile-compilation-
dir -ffile-prefix-map -ffile-reproducible -ffine-grained-bitfield-accesses -ffinite-loops -ffinite-math-only -ffixed-a0 -ffixed-a1 -ffixed-a2 -ffixed-a3 -ffixed-a4 -ffixed-a5 -ffixed-a6 -ffixed-a6 -ffixed-d0 -ffixed-d1 -ffixed-d2
-ffixed-d3 -ffixed-d4 -ffixed-d5 -ffixed-d6 -ffixed-d7 -ffixed-point -ffixed-r19 -ffixed-r9 -ffixed-x10 -ffixed-x11 -ffixed-x12 -ffixed-x13 -ffixed-x14 -ffixed-x15 -ffixed-x16 -ffixed-x17 -ffixed-x18 -ffixed-x19
-ffixed-x1 -ffixed-x20 -ffixed-x21 -ffixed-x22 -ffixed-x23 -ffixed-x24 -ffixed-x25 -ffixed-x26 -ffixed-x27 -ffixed-x28 -ffixed-x29 -ffixed-x2 -ffixed-x30 -ffixed-x31 -ffixed-x3 -ffixed-x4 -ffixed-x5 -ffixed-x6 -ffixed-
x7 -ffixed-x8 -ffixed-x9 -fforce-dwarf-form -fforce-emit-vtables -fforce-enable-int128 -ffp-contract -ffp-eval-method -ffp-exception-behavior -ffp-model -ffreestanding -ffuchsia-api-level -ffunction-sections
-fgeneric-block-helpers -fglobal-isel -fgnu-keywords -fgnu-runtime -fgnu89-inline -fgnu-version -fgpu-allow-device-init -fgpu-default-stream -fgpu-defer-diag -fgpu-flush-denormals-to-zero -fgpu-rdc -fgpu-sanitize
-fhip-emit-relocatable -fhip-fp32-correctly-rounded-divide -fhip-kernel-arg-name -fhip-new-launch-api -fhip-non-infinities -fhip-non-nans -fignore-exceptions -fimplicit-module-maps -fin-descriptor-block-flags
-fincremental-extensions -finline-functions -finline-hint-functions -finline-max-stacksize -finput-charset -finstrument-function-entry-bar -finstrument-functions-after-inlining -finstrument-functions -fintegrated-as
-fintegrated-cc1 -fintegrated-objemitter -fjmc -fjump-tables -fkeep-persistent-storage-variables -fkeep-static-consts -flax-vector-conversions -flto-jobs -fltoauto -fltojobserver -flto -fmacro-backtrace-limit -fmacro-
prefix-map -fmath-errno -fmax-tokens -fmax-type-align -fmemory-profile-use -fmemory-profile -fmemory-profile -fmerge-all-constants -fmmessage-length -fminimize-whitespace -fmodule-file[] -fmodule-header -fmodule-header
-fmodule-map file -fmodule-name -fmodule-output -fmodule-output -fmodule-related-to-pch -fmodulemap-allow-subdirectory-search -fmodules-cache-path -fmodules-decluse -fmodules-disable-diagnostic-validation -fmodules-
ignore-macro -fmodules-prune-after -fmodules-prune-interval -fmodules-search-all -fmodules-skip-diagnostic-options -fmodules-skip-header-search-paths -fmodules-strict-decluse -fmodules-user-build-path -fmodules-
validate-input-files-content -fmodules-validate-once-per-build -fmodules-validate-system-headers -fmodules -fms-compatibility-version -fms-compatibility -fms-extensions -fms-hotpatch -fms-runtime-lib -fmsc-version
-fnew-alignment -fnew-infallible -fno-aapcs-bitfield-width -fno-access-control -fno-addrsig -fno-apinotes-modules -fno-apinotes -fno-assume-sane-operator-new -fno-assume-unique-vtables -fno-autolink -fno-builtin -fno-
c++-static-constructors -fno-char8_t -fno-color-diagnostics -fno-common -fno-complete-member-pointers -fno-constant-cfstrings -fno-constant-nsarray-literals -fno-constant-nsdictionary-literals -fno-constant-nsnumber-
literals -fno-convergent-functions -fno-coverage-mapping -fno-crash-diagnostics -fno-cuda-approx-transcendentals -fno-cxx-modules -fno-debug-macro -fno-declspec -fno-define-target-os-macros -fno-delayed-template-
parsing -fno-delete-null-pointer-checks -fno-depscan-share -fno-diagnostics-fixit-info -fno-diagnostics-show-line-numbers -fno-digraphs -fno-direct-access-external-data -fno-discard-value-names -fno-dollars-in-
identifiers -fno-elide-constructors -fno-elide-type -fno-eliminate-unused-debug-types -fno-exceptions -fno-experimental-relative-c++-abi -fno-experimental-sanitize-metadata -fno-file-reproducible -fno-fine-grained-
bitfield-accesses -fno-finite-loops -fno-fixed-point -fno-force-enable-int128 -fno-generic-block-helpers -fno-global-isel -fno-gnu-inline-asm -fno-gpu-allow-device-init -fno-gpu-defer-diag -fno-hip-emit-relocatable
-fno-hip-fp32-correctly-rounded -fno-hip-kernel-arg-name -fno-hip-new-launch-api -fno-in-descriptor-block-flags -fno-integrated-as -fno-integrated-cc1 -fno-integrated-objemitter -fno-jump-tables -fno-keep-persistent-
storage-variables -fno-keep-static-consts -fno-krn-functions -fno-lto -fno-memory-profile -fno-merge-all-constants -fno-modules-check-relocated -fno-modules-prune-non-affecting -fno-new-infallible -fno-objc-infer-
related-result -fno-offload-lto -fno-openmp-extensions -fno-operator-names -fno-optimize-sibling-calls -fno-pch-codegen -fno-pch-codegen -fno-plt -fno-preserve-as-comments -fno-profile-generate -fno-profile-instr-
generate -fno-profile-instr-use -fno-pseudo-probe-for-profiling -fno-ptrauth-abi-version -fno-ptrauth-kernel-abi-version -fno-register-global-dtors-with -fno-rtlib-add-rpath -fno-rtti-data -fno-rtti -fno-sanitize-
address-globals-dead -fno-sanitize-address-outline-instrumentation -fno-sanitize-address-poison-custom -fno-sanitize-address-use-after -fno-sanitize-address-use-odr -fno-sanitize-cfi-canonical-jump -fno-sanitize-cfi-
cross-dso -fno-sanitize-coverage -fno-sanitize-hwaddress-experimental-aliasing -fno-sanitize-ignoreslist -fno-sanitize-memory-param-retval -fno-sanitize-memory-track-origins -fno-sanitize-memory-use-after -fno-sanitize-
recover -fno-sanitize-stable-abi -fno-sanitize-stats -fno-sanitize-thread-atomics -fno-sanitize-thread-func-entry -fno-sanitize-thread-memory-access -fno-sanitize-thread-memory-access -fno-sanitize-trap -fno-sanitize-trap -fno-short-wchar -fno-show-
column -fno-show-source-location -fno-signed-char -fno-signed-zeros -fno-spell-checking -fno-split-machine-functions -fno-split-stack -fno-stack-check -fno-stack-clash-protection -fno-stack-protector -fno-standalone-
debug -fno-strict-float-cast-overflow -fno-strict-return -fno-sycl -fno-temp-file -fno-threads-safe-statics -fno-trap-function-returns -fno-trigraphs -fno-typed-cxx-delete -fno-typed-cxx-new-delete -fno-typed-memory-
operations-experimental -fno-unified-lto -fno-unique-section-names -fno-unroll-loops -fno-use-cxa-atexit -fno-use-init-array -fno-visibility-inlines-hidden-static -fno-xray-function-index -fno-zero-initialized-in-bss
-fobjc-arc-exceptions -fobjc-arc -fobjc-disable-direct-methods-for -fobjc-encode-cxx-class-template -fobjc-exceptions -fobjc-relative-method-lists -fobjc-runtime -fobjc-weak -foffload-lto -foffload-lto -fomit-frame-
pointer -fopenmp-extensions -fopenmp-offload-mandatory -fopenmp-simd -fopenmp-target-debug -fopenmp-target-jit -fopenmp-targets -fopenmp-version -fopenmp -fopenter-arrow-depth -foptimization-record-file
-foptimization-record-passes -forder-file-instrumentation -fpack-struct -fpascal-strings -fpascal-plugin -fpatchable-function-entry> -fpcc-struct-return -fpch-codegen -fpch-debuginfo -fpch-instantiate-templates -fpch-
validate-input-files-content -fplugin-arg- -fplugin -fprebuilt-implicit-modules -fprebuilt-module-path -fproc-stat-report -fproc-stat-report -fprofile-arcs -fprofile-exclude-files -fprofile-filter-files -fprofile-
function-groups -fprofile-generate -fprofile-generate -fprofile-instr-generate -fprofile-instr-generate -fprofile-instr-use -fprofile-list -fprofile-remapping-file -fprofile-sample-accurate -fprofile-sample-use
-fprofile-selected-function-group -fprofile-update -fprofile-use -fprotect-parens -fpseudo-probe-for-profiling -fptrauth-abi-version -fptrauth-abi-version -fptrauth-auth-traps -fptrauth-block-descriptor-pointers -fptrauth-calls -fptrauth-
function-pointer-type-discrimination -fptrauth-indirect-gotos -fptrauth-intrinsics -fptrauth-kernel-abi-version -fptrauth-objc-class-rod -fptrauth-objc-interface-sel -fptrauth-objc-isa-masking -fptrauth-objc-isa-mode
-fptrauth-objc-isa -fptrauth-returns -fptrauth-soft -fptrauth-type-info-discriminated-vtable -fptrauth-vtable-pointer-address-discrimination -fptrauth-vtable-pointer-type-discrimination -frandomize-layout-seed-file
-frandomize-layout-seed -freciprocal-math -freg-struct-return -fregister-global-dtors-with-atexit -frelaxed-template-template-args -freroll-loops -frpfi -frtlib-add-rpath -frwpi -fsafe-buffer-usage-suggestions
-fsample-profile-use-profi -fsanitize-address-destroyer -fsanitize-address-field-padding -fsanitize-address-globals-dead-stripping -fsanitize-address-outline-instrumentation -fsanitize-address-poison-custom-array
-fsanitize-address-use-after-return -fsanitize-address-use-after-scope -fsanitize-address-use-odr-indicator -fsanitize-blacklist -fsanitize-cfi-canonical-jump-tables -fsanitize-cfi-cross-dso -fsanitize-cfi-icall-
experimental-normalize -fsanitize-cfi-icall-generalize-pointers -fsanitize-coverage-allowlist -fsanitize-coverage-ignoreslist -fsanitize-coverage -fsanitize-hwaddress-abi -fsanitize-hwaddress-experimental-aliasing
-fsanitize-ignoreslist -fsanitize-memory-param-retval -fsanitize-memory-track-origins -fsanitize-memory-track-origins -fsanitize-memory-use-after-dtor -fsanitize-memtag-mode -fsanitize-recover -fsanitize-stable-abi
-fsanitize-stats -fsanitize-system-ignoreslist -fsanitize-thread-atomics -fsanitize-thread-func-entry-exit -fsanitize-thread-memory-access -fsanitize-trap -fsanitize-trap -fsanitize-undefined-strip-path-components
-fsanitize -fsave-optimization-record -fsave-optimization-record -fseh-exceptions -fshort-enums -fshort-wchar

```

cuda-ptxas -Xlinker -Xoffload-linker -Xopenmp-target -Xopenmp-target -Xpreprocessor -x -z -fshow-overloads -fshow-skipped-includes -fsigned-char -fsized-deallocation -fsjlj-exceptions -fslp-vectorize -fsmall-block-descriptors -fspell-checking-limit -fsplit-dwarf-inlining -fsplit-lto-unit -fsplit-machine-functions -fsplit-stack -fstack-check -fstack-clash-protection -fstack-protector-all -fstack-protector-strong -fstack-protector -fstack-size-section -fstack-usage -fstandalone-debug -fstrict-enums -fstrict-flex-arrays -fstrict-float-cast-overflow -fstrict-vtable-pointers -fsuppress-conflicting-types -fswift-async-fp -fsycl -fsyntax-only -fsystem-module -ftarget-variant-availability-checks -ftemplate-backtrace-limit -ftemplate-depth -ftest-coverage -fthin-link-bitcode -fthinlto-index -ftime-report -ftime-trace-granularity -ftime-trace -ftime-trace -ftrap-function-returns -ftrap-function -ftrapv-handler -ftrapv -ftrigraphs -ftrivial-auto-var-init-skip -ftrivial-auto-var-init-skip -ftrivial-auto-var-init-skip -ftrivial-auto-var-init-skip -ftrivial-auto-var-init-stop -ftrivial-auto-var-init -ftrivial-auto-var-zero-init -ftyped-cxx-delete -ftyped-cxx-new-delete -ftyped-memory-operations-experimental -funifed-lto -funique-basic-block-section-names -funique-internal-linkage-names -funroll-loops -funsafe-math-optimizations -fuse-cuid -fuse-line-directives -fvalidate-ast-input-files-content -fvecLib -fvectortize -fverbose-asm -fvirtual-function-elimination -fvisibility-dllexport -fvisibility-externs-dllimport -fvisibility-externs-nodllstorageclass -fvisibility-from-dllstorageclass -fvisibility-global-new-delete-hidden -fvisibility-inlines-hidden-static-local -fvisibility-inlines-hidden -fvisibility-ms-compat -fvisibility-nodllstorageclass -fvisibility-fwasm-exceptions -fwhole-program-vtables -fwrap -fwritable-strings -fxl-pragma-pack -fxray-always-emit-customevents -fxray-always-emit-typedevents -fxray-always-instrument -fxray-attr-list -fxray-function-groups -fxray-ignore-loops -fxray-instruction-threshold -fxray-instrumentation-bundle -fxray-instrument -fxray-link-deps -fxray-modes -fxray-never-instrument -fxray-selected-function-group -fzero-call-used-regs -fzvector -F --gcc-install-dir --gcc-toolchain -gcodeview-command-line -gcodeview-ghash -gcodeview-gdwarf-2 -gdwarf-3 -gdwarf-4 -gdwarf-5 -gdwarf32 -gdwarf64 -gdwarf -gembed-source -gen-reproducer -gline-directives-only -gline-tables-only -gmodules -gno-codeview-command-line -gno-embed-source -gno-inline-line-tables -gpu-bundle-output -gpu-instrument-lib -gpu-max-threads-per-block -greproducible -gsplit-dwarf -gstrict-dwarf -gz -G -g -help-hidden -help --hip-device-lib --hip-link --hip-path --hip-version --hipspv-pass-plugin -H -I -iapi-notes-modules -iapi-notes-path -ibuiltinc -idirafter -iframeworkwithsysroot -iframework -imacros -include-pch -include -index-header-map -index-ignore-macros -index-ignore-pcms -index-ignore-system-symbols -index-record-codegen-name -index-store-path -index-unit-output-path -iprefix -iquote -isysroot -isystem-after -isystem -ivfsoverlay -ivfstatscache -iwithprefixbefore -iwithprefix -iwithsysroot -I --libomptarget-amdgcn-bc-path --libomptarget-amdgpu-bc-path --libomptarget-nvptx-bc-path -L -mabiquadword-atomics -mabiccalls -maix-struct-return -malign-branch-boundary -malign-branch -malign-double -maltivec -mamdgpu-ieee -march -mbackchain -mbranch-protection -mbranches-within-32B-boundaries -mcabac -mcmodelmedany -mcmodelmedlow -mcmse -mcode-object-version -mcpu -mcrcbits -mcrc -mcumode -mdefault-visibility-export-mapping -mdouble -MD -meabi -membedded-data -menable-experimental-extensions -mexec-model -mexecute-only -mextern-sdata -mfentry -mfix-cmse-cve-2021-35465 -mfix-cortex-a53-835769 -mfix-cortex-a57-aes-1742098 -mfix-cortex-a72-aes-1655431 -mfp32 -mfp64 -mframe-chain -mfunction-return -MF -mgeneral-regs-only -mglobal-merge -mgpopt -mguard -MG -mharden-sls -mhvx-ieee-fp -mhvx-length -mhvx-qfloat -mhvx -mhvx -miamcu -mignore-ccoff-visibility -migrate -mincremental-linker-compatible -mindirect-branch-cs-prefix -mindirect-jump -mios-version-min -MJ -mllvm -mllvm -mlocal-sdata -mlong-calls -mlong-double-128 -mlong-double-64 -mlong-double-80 -mlvi-cfi -mlvi-hardening -mmacos-version-min -mmadd4 -mmark-bti-property -MMD -mmemops -mmir -mms-bitfields -mmsa -mmt -MM -mno-abiccalls -mno-bti-at-return-twice -mno-crc -mno-cumode -mno-embedded-data -mno-execute-only -mno-extern-sdata -mno-fix-cmse-cve-2021-mno-fix-cortex-a53-835769 -mno-fix-cortex-a57-aes -mno-fix-cortex-a72-aes -mno-fmv -mno-gather -mno-global-merge -mno-gpopt -mno-hvx-ieee-fp -mno-hvx-qfloat -mno-hvx -mno-implicit-float -mno-implicit-sme -mno-incremental-linker-compatible -mno-local-sdata -mno-long-calls -mno-lvi-cfi -mno-lvi-hardening -mno-madd4 -mno-memops -mno-movt -mno-ms-bitfields -mno-msa -mno-mt -mno-neg-immediates -mno-nvj -mno-nvs -mno-outline-atomics -mno-outline -mno-packets -mno-pic-data-is-text -mno-relax -mno-restrict-it -mno-save-restore -mno-scatter -mno-ses -mno-stack-arg-probe -mno-tgsplit -mno-tls-direct-seg-refs -mno-unaligned-access -mno-wavefrontsize64 -mnocrc -mno-pcount -mnvj -mnvs -module-dependency-dir -module-file-info -momit-leaf-frame-pointer -moutline-atomics -moutline -mpacked-stack -mpackets -mpad-max-prefix-size -mpic-data-is-text -relative -mprefer-vector-width -mprintf-kind -MP -mqdsp6-compat -MQ -mrecip -mrecip -mrecord-mcount -mrelax-all -mrelax -mrestrict-it -mrtld -mrvv-vector-bits -msave-restore -mses -msign-return-address -mskip-rax-setup -msmall-data-limit -msoft-float -mstack-alignment -mstack-arg-probe -mstack-probe-size -mstack-protector-guard-offset -mstack-protector-guard-reg -mstack-protector-guard-symbol -mstack-protector-guard -mstackrealign -msve-vector-bits -msvr4-struct-return -mtargetos -mtgsplit -mthread-model -mtls-direct-seg-refs -mtls-size -mtp -mtune -MT -munaligned-access -munsafe-fp-atomics -mvscale-max -mvscale-min -MV -mwavefrontsize64 -mxcoff-build-id -mxcoff-roptr -mzos-hlq-clang -mzos-hlq-csslib -mzos-hlq -mzos-sys-include -M --no-cuda-include-ptx --no-cuda-version-check --no-default-config --no-gpu-bundle-output -no-hip-rt --no-offload-arch --no-offload-new-driver --no-system-header-prefix -nobuiltinc -nogpuinc -nogpulib -nohipwrapperinc -nostdinc++ --nvptx-arch-tool -ObjC++ -objcmt-allowlist-dir-path -objcmt-atomic-property -objcmt-migrate-all -objcmt-migrate-annotation -objcmt-migrate-designated-init -objcmt-migrate-instancetype -objcmt-migrate-literals -objcmt-migrate-macros -objcmt-migrate-property-dot-syntax -objcmt-migrate-property -objcmt-migrate-protocol-conformance -objcmt-migrate-readonly-property -objcmt-migrate-readwrite-property -objcmt-migrate-subscripting -objcmt-ns-nonatomic-iosonly -objcmt-returns-innerpointer-property -objcmt-whitelist-dir-path -ObjC -object-file-name -offload-arch --offload-device-only --offload-host-device --offload-host-only --offload-link --offload-new-driver --offload -o -pedantic -pg -pipe --precompile --pretty-sgf -print-diagnostic-options -print-effective-triple -print-file-name -print-ivar-layout -print-libgcc-file-name -print-multi-flags-experimental -print-program-name -print-resource-dir -print-roc-cc-search-dirs -print-runtime-dir -print-search-dirs -print-supported-cpus -print-target-triple -print-targets -pthread -ptxas-path -P -p -Qn -Qunused-arguments -Qy -relocatable-pch -rewrite-legacy-objc -rewrite-objc --roc-cc-device-lib-path --roc-cc-path -Rpass-analysis -Rpass-missed -Rpass -rtlib -R -save-stats -save-stats -save-temps -save-temps -serialize-diagnostics -shared-libs -start-no-unused-arguments -static-libs -static-libsan -static-openmp -std -stdlib++-isystem -stdlib -sycl-std --symbol-graph-dir --system-header-prefix -S --target -time -traditional-cpp -trigraphs -T -undef -unwindlib -U --verify-debug-info -verify-pch --version -vfsoverlay -v -Wa -Wdeprecated -Wl -working-directory -Wp -W -w -Xanalyzer -Xarch_device -Xarch_host -Xassembler -Xclang -Xclang -Xcuda-f

Problems Meson solves:

- Manage compilation of libraries, and any libraries used by them
 - Locate libraries installed on your system through other means
 - Libraries can create executables of their own
 - Portions of libraries can be enabled or disabled
- Track which compiler flags are needed to compile each file

Problems Meson solves:

- Manage compilation of libraries, and any libraries used by them
 - Locate libraries installed on your system through other means
 - Libraries can create executables of their own
 - Portions of libraries can be enabled or disabled
 - Track which compiler flags are needed to compile each file
 - Libraries can change behaviour depending on whether a specific library or program is available on your computer
 - Libraries sometimes generate source code that depends on all points listed above
- Handle source files written in different languages (e.g. C, CUDA)
- Ability to produce 'recipe files' for different build tools (e.g. Visual Studio, Ninja, Make)

Libraries

- The fastest way to get from A to B is usually using a library of some kind
- Being familiar with some libraries can save you a lot of time in the long run!
- Or make the language more convenient to use

A hitchhiker's guide to libraries

- I have found a library I want to use in my project.

What do I do?

A hitchhiker's guide to libraries

- Is the library a header-only library?
 - This is the easiest kind to work with. You just drop it in your source directory and you're good to go

A library's description will usually tell you if it is

Arrrgh

Jeff Wofford | July 2014 | e: biz@jeffwofford.com | w: <http://www.jeffwofford.com>


arrrgh is a fast, small, simple, powerful **single-header library** for parsing command line arguments in C++ using more-or-less POSIX parsing rules. It is written using modern C++11 paradigms so it's safe, fast, and tends to invite clean, small, to-the-point code.

There are lots of other command line parsers out there. Yet every time I wrote a new command line program, I ended up rolling my own underpowered and error-prone argument parsing. I like

A hitchhiker's guide to libraries

- Is the library a header-only library?
 - This is the easiest kind to work with. You just drop it in your source directory and you're good to go
- Alternatively: you can add the directory to the locations where the compiler looks for header files in your meson.build file

```
exe = executable('program', src, 'main.cpp',  
    include_directories: ['path/to/library'],  
    dependencies : [animationwindow_dep, sdl2_dep, std_lib_facilities_dep],  
    cpp_args : compiler_flags  
)
```



Intermezzo - Task:

How can a header-only library exist in C++?

Review question:

- What can happen during the compilation of a program when we place function definitions in header files (.h), instead of C++ source files (.cpp)?

Task:

- Name a C++ language feature that would allow the creation of a header-only library (all function definitions are contained in the header file)

Note: there are several correct answers here, but only one is a part of the syllabus

A hitchhiker's guide to libraries

- Is the library a header-only library?
- If not: is it available on wrapdb?



<https://github.com/mesonbuild/wrapdb/tree/master/subprojects>

A hitchhiker's guide to libraries

- Is the library a header-only library?
- If not: is it available on wrapdb?
 - If so, download the .wrap file, and place it in the 'subprojects' directory. The latter must be in the same directory as your meson.build file.
 - Meson mandates this name and structure
- Add it as a dependency in your meson.build file:

```
library_dep = dependency('librarynamegoeshere')
```
- Then add it as a dependency to your executable:

```
dependencies: [library_dep, animationwindow_dep, sdl2_dep]
```



A hitchhiker's guide to libraries

- Is the library a header-only library?
- If not: is it available on wrapdb?
- Still no? Do you see a CMakeLists.txt file?
 - The project uses cmake as its build system, which is practically the industry standard
- Place the library in the subprojects directory
- Then add it in your meson.build file:

```
cmake = import('cmake')
library_project = cmake.subproject('library_to_add')
library_dep = library_project.dependency('usuallythelibraryname')

exe = executable('program', src,
    dependencies : [library_dep, animationwindow_dep, sdl2_dep],
)
```


A hitchhiker's guide to libraries

- Is the library a header-only library?
- If not: is it available on wrapdb?
- Still no? Do you see a CMakeLists.txt file?
- Absolutely none of the above?
- Usually requires jumping through hoops
 - But also uncommon!
 - In my experience: most commonly libraries that are compiled using makefiles
- Is another library available?

Makefile

- A text file that is called «Makefile»
- Tries to be a smart shell script
- Can do partial compilation
- Syntax is often unintuitive

```
BUILDDIR = build  
EXECUTABLE = $(BUILDDIR)/program
```

```
CXX = clang++  
CXXFLAGS = -std=c++17  
LINK = $(CXX)  
LDFLAGS = -o $(EXECUTABLE)
```

```
SOURCES := $(wildcard *.cpp)  
SOURCES_OBJECTS = $(patsubst %.cpp,%.o,$(SOURCES:.cpp=.o))
```

```
SRC=${wildcard *.cpp}  
OBJ=${patsubst %.cpp,build/%.o,${SRC}}
```

```
all: ${OBJ}  
    ${CXX} ${CFLAGS} ${OBJ} ${LDFLAGS}
```

```
$(BUILDDIR)/%.o: %.cpp  
    mkdir -p ${dir $@}  
    ${CXX} -o $@ $< -c ${CFLAGS}
```

```
clean:  
    rm -rf $(BUILDDIR)
```

Today

- Libraries and the Meson build system
- **How to develop a program**

Tips for developing a program

- Tip 1: Start with your data
 - Create classes and other data types that your program will need to use
 - Usually dictates to some degree how your program needs to interact with these classes.
 - Example:

```
enum class ChessPiece {NONE, PAWN, ROOK, KNIGHT,  
    BISHOP, QUEEN, KING};  
  
class ChessBoard {  
    std::array<std::array<ChessPiece, 8>, 8> board;  
};
```

Tips for developing a program

- Tip 2: Don't try to plan too much
 - You cannot really foresee all complexities in your program until you encounter them while coding
- Create something that works first, then rework it as needed.

Today

- Libraries and the Meson build system
- How to develop a program
- **The main() function**

The main() function

- The long variant of main() allows taking in command line parameters:

```
int main(int argc, char** argv) {  
    return 0;  
}
```

- How to interpret these:
 - argc: The number of command line arguments given
 - argv: A C-style array containing C-style strings (char*)

The main() function

- We recommend one of two strategies.

Strategy 1: convert to vector of strings

```
int main(int argc, char** argv) {  
    std::vector<std::string> arguments;  
    for(int i = 0; i < argc; i++) {  
        arguments.push_back(argv[i]);  
    }  
  
    // Use arguments here  
}
```

Strategy 2: use a library for interpreting arguments

My personal library of choice: <https://github.com/ElectricToy/arrngh>

The `main()` function

- The first argument is always the path to your program executable file.
 - Thus, `argc` is always at least 1.
- Opening a file with a program places it as its second command line parameter with index 1 (because of the point above).
- Similarly, if you open n files with your program, you will end up with $n + 1$ program arguments

Today

- Libraries and the Meson build system
- How to develop a program
- The main() function
- **The C++ standard library**

The Standard Template Library (STL)

- We have already used a bunch of it:
 <vector> <array> <map> <unordered_map>
 <string> <filesystem> <iostream> <fstream>
 <random>
- We'll look at some more useful bits now:
 - Containers
 - Operations on containers
 - Some math functions
 - The print() function

The Standard Template Library (STL)

- We'll look at some more useful bits now:
 - **Containers**
 - Operations on containers
 - Some math functions
 - The `print()` function

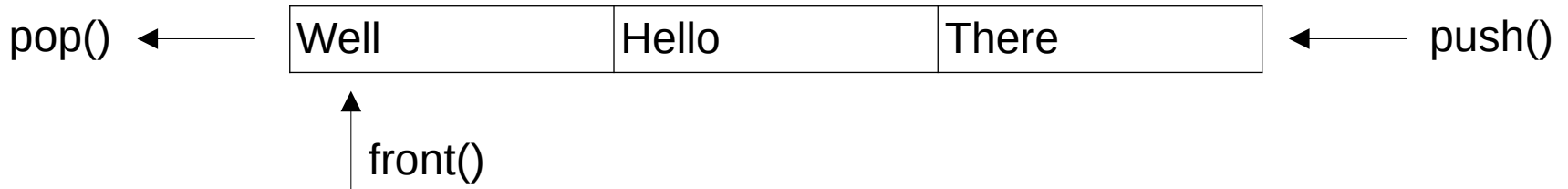
Containers

- `std::queue`

```
std::queue<std::string> messagesQueue;
```

```
messagesQueue.push("Well");  
messagesQueue.push("Hello");  
messagesQueue.push("There");
```

```
while(!messagesQueue.empty()) {  
    std::cout << "Entry in queue: " << messagesQueue.front() << std::endl;  
    messagesQueue.pop();  
}
```



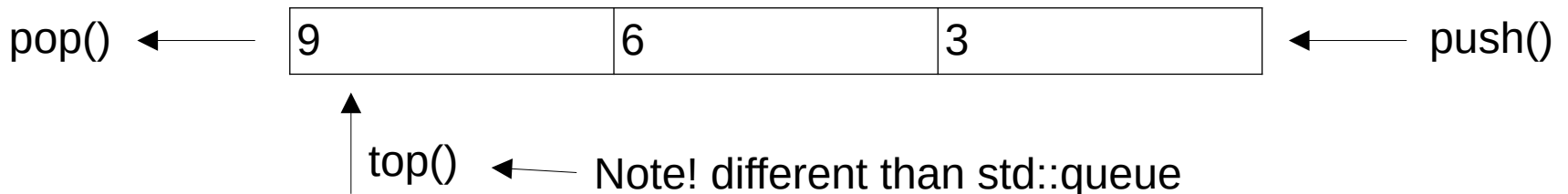
Containers

- `std::priority_queue`: picks the highest value element using the `<` operator (overload it to use a class)

```
std::priority_queue<int> messagesQueue;
```

```
messagesQueue.push(3);  
messagesQueue.push(9);  
messagesQueue.push(6);
```

```
while(!messagesQueue.empty()) {  
    std::cout << "Entry in queue: " << messagesQueue.top() << std::endl;  
    messagesQueue.pop();  
}
```



Containers

- `std::set`: used to see if you have seen a value before (overload `operator<` if you want to use your own class)

```
struct Number {  
    int value = 0;  
    bool operator<(const Number& other) const {  
        return value < other.value;  
    }  
};  
  
int main() {  
    Number a{5}; Number b{3}; Number c{5};  
    std::set<Number> numbers;  
    numbers.insert(a); numbers.insert(b);  
    std::cout << numbers.contains(c) << std::endl; // true  
}
```

The Standard Template Library (STL)

- We'll look at some more useful bits now:
 - Containers
 - **Operations on containers**
 - Some math functions
 - The print() function

Container operations

- Practically all containers have a `begin()` and `end()` function. This allows them to be used with most™ algorithms in the standard library

- Sort a list:

```
std::vector<int> numberList = {2, 5, 3, 7, 5, 9};  
std::sort(numberList.begin(), numberList.end());
```

- Copy a list:

```
// Initialise to length of 20  
std::vector<int> largerList(20);  
std::copy(numberList.begin(), numberList.end(),  
          // Copy to index 3 and onwards  
          largerList.begin() + 3);
```

Container operations

- Fill a list with the same value:

```
std::vector<int> numberList = {2, 5, 3, 7, 5, 9};  
std::fill(numberList.begin(), numberList.end(), 9);  
// Result: 9, 9, 9, 9, 9, 9
```

- Fill a list with increments of a value (using the ++ operator):

```
// Initialise to length of 20  
std::vector<int> oneToTwenty(20);  
std::iota(oneToTwenty.begin(), oneToTwenty.end(), 1);  
// Result: 1, 2, 3, 4, ..., 19, 20
```

Container operations

- Randomly shuffle the elements of a container:

```
std::vector<int> numbers {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
std::random_device rd;  
std::default_random_engine engine(rd());  
std::shuffle(numbers.begin(), numbers.end(), engine);
```

- Reverse the order of the elements in a container:

```
std::reverse(numbers.begin(), numbers.end());
```

The Standard Template Library (STL)

- We'll look at some more useful bits now:
 - Containers
 - Operations on containers
 - **Some math functions**
 - The `print()` function

The Standard Template Library (STL)

- Some math functions:

```
std::max<float>(1.0, 2.0); // returns 2.0  
std::min<int>(-3, 5); // returns -3
```

```
float lowerLimit = 0;  
float upperLimit = 1;  
// returns 1.0  
std::clamp<float>(1.2, lowerLimit, upperLimit);  
  
std::abs(-1); // returns 1
```

The Standard Template Library (STL)

- Some math functions:

```
std::log()  
std::sqrt()  
std::cos()  
std::sin()
```

```
float value = 1.3;  
std::floor(value); // returns 1.0  
std::ceil(value);  // returns 2.0  
std::round(value); // returns 1.0
```

The Standard Template Library (STL)

- We'll look at some more useful bits now:
 - Containers
 - Operations on containers
 - Some math functions
 - **The print() function**

std::print() and std::println()

- Nicer way to print to the terminal
- Available with newer revisions of the standard library (C++23)
 - But not those used in the course
 - Very similar to the fmt library, should use that instead for the moment

```
#include <print>
```

```
int main() {  
    std::println("{2} {1} {0}!", 23, "from C++", "Only available");  
  
    return 0;  
}
```


Today

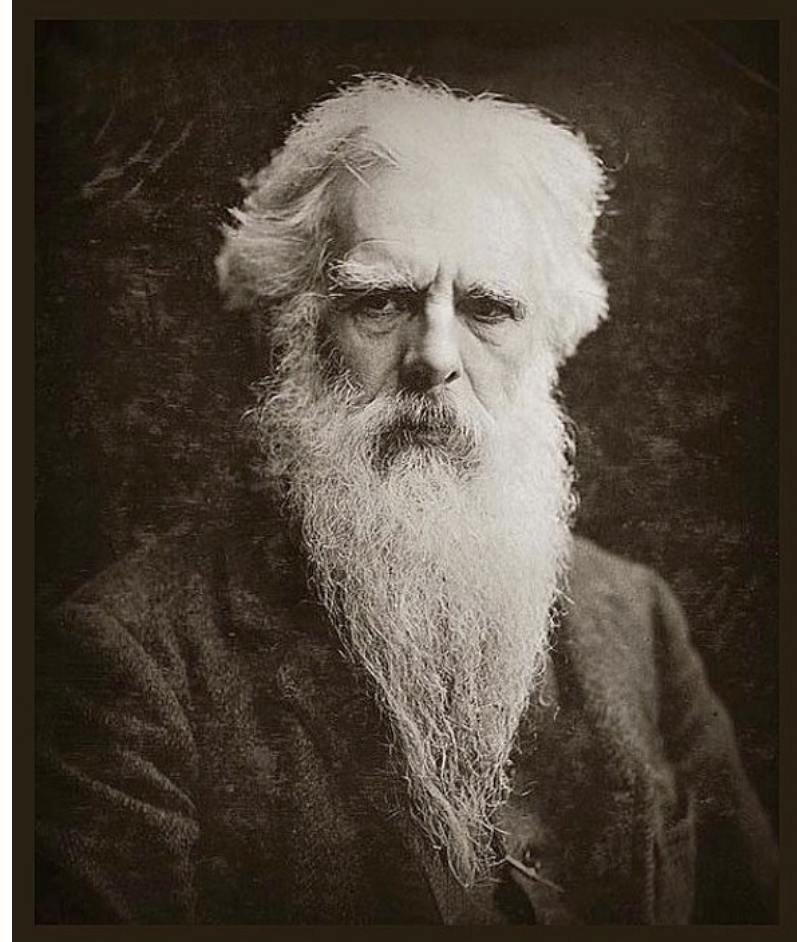
- Libraries and the Meson build system
- How to develop a program
- The main() function
- The C++ standard library
- **Animation**

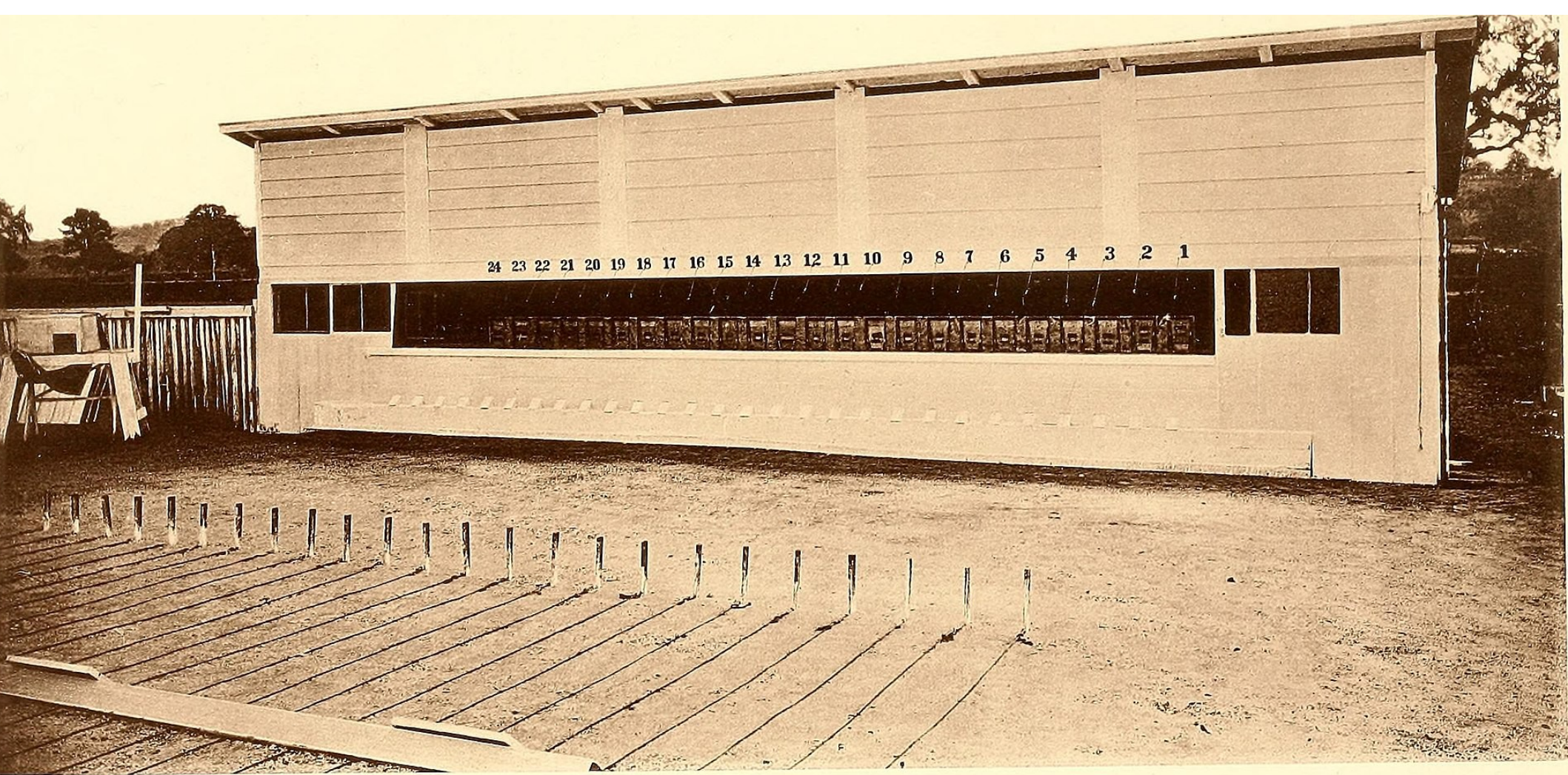
1878

Leland Stanford

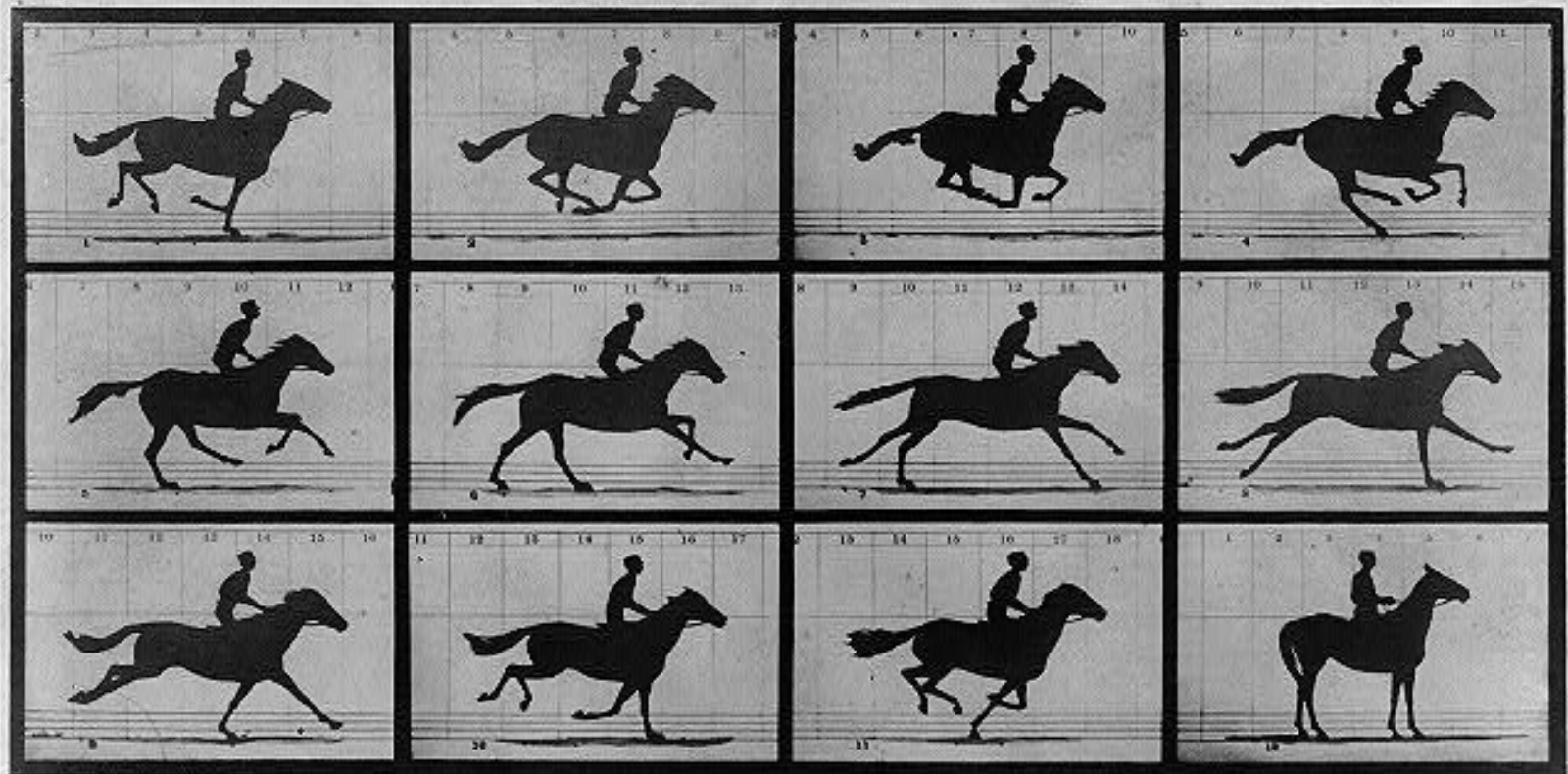


Eadweart Muybridge





May? Yes of course you may! <3



Copyright, 1878, by MUYBRIDGE.

MORSE'S Gallery, 417 Montgomery St., San Francisco.

THE HORSE IN MOTION.

Illustrated by
MUYBRIDGE.

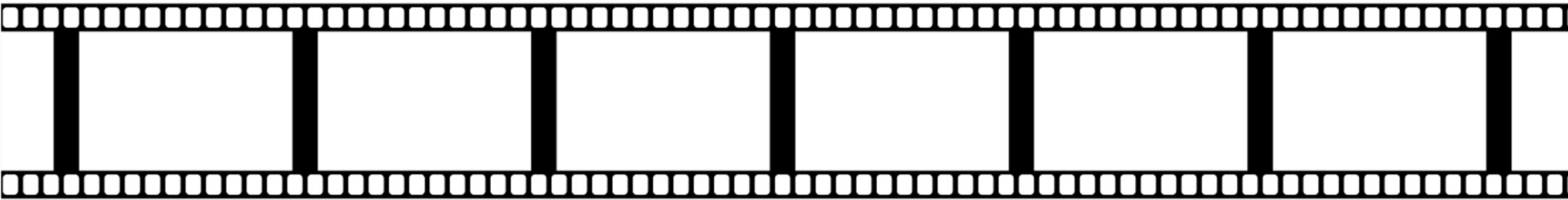
AUTOMATIC ELECTRO-PHOTOGRAPH.

"SALLIE GARDNER," owned by LELAND STANFORD; running at a 1.40 gait over the Palo Alto track, 19th June, 1878.

The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed in each twenty-seven inches of progress during a single stride of the horse. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each. The exposure of each negative was less than the two-thousandth part of a second.

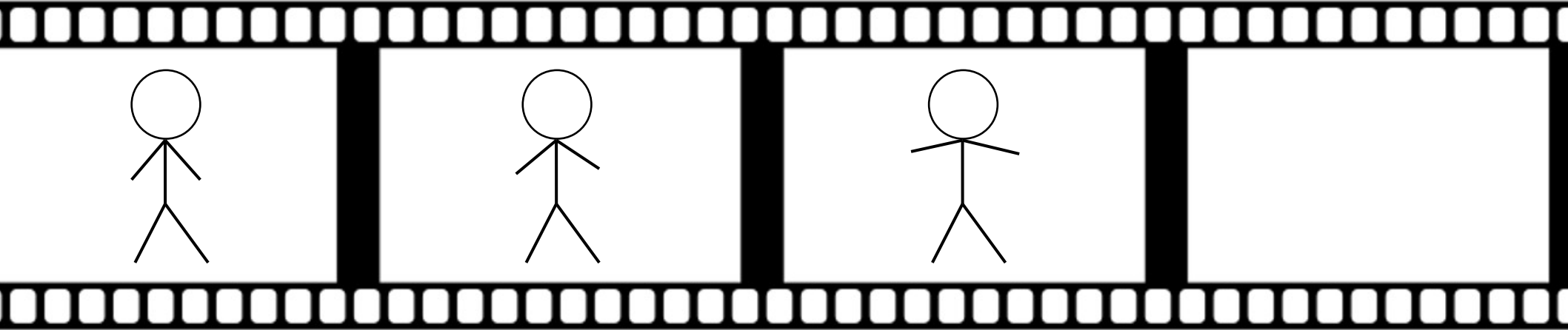
May? Yes of course you may! <3

Let's make

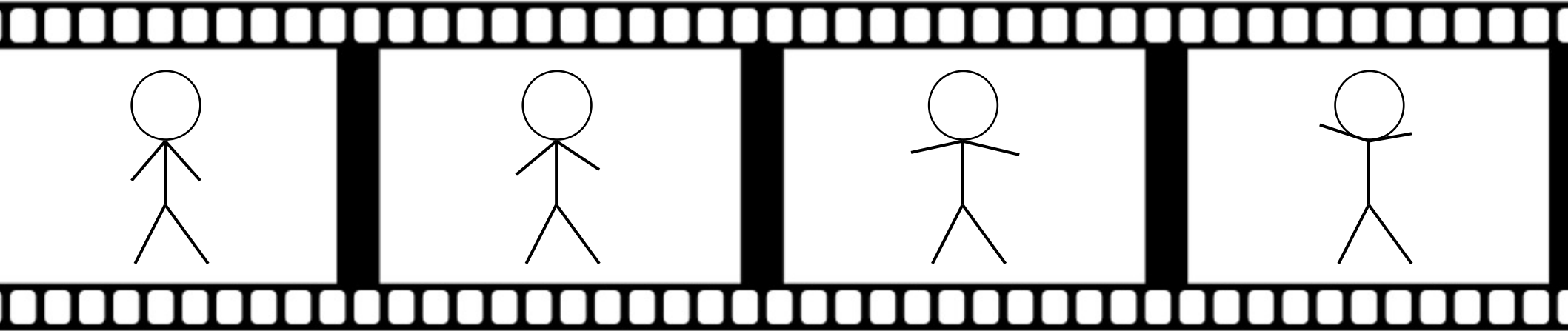


an animation

1. Create a new empty frame



1. Create a new empty frame



2. Draw a picture in that frame

Animation

```
AnimationWindow window;
int xCoordinate = 0;
while(!window.should_close()) {
    xCoordinate++;
    if(xCoordinate > 500) {
        xCoordinate = 0;
    }
    window.draw_circle({xCoordinate, 100}, 50);
    window.next_frame();
}
```

Keep running until the window is closed

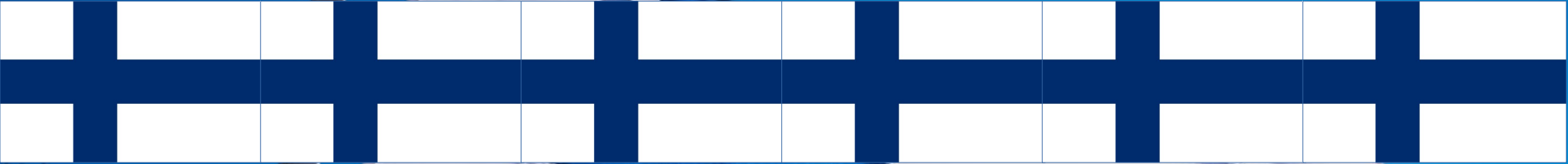
To animate something, we just have to move it a little bit for each picture we draw

We have drawn a picture, and can move on to drawing the next one

We have made it..

To the end of the syllabus!

And this slide marks the Finnish line!



Demonstration: useful libraries

- `nlohmann::json`
 - Used for reading, editing, and writing JSON.
 - JSON is a structured data format that is widely supported.
- STB libraries
 - Public domain single header libraries
- `Tabulate`
 - Create nice looking tables in the terminal
- `Arrrgh`
 - Command line option parser
- `FMT`
 - Fast and versatile string printing and formatting library

Today

- Libraries and the Meson build system
- How to develop a program
- The main() function
- The C++ standard library
- Animation

Next week

- Going over several topics once more
 - Remember to let me know which topics you want to see!

