Latches and Barriers

A thread waits at a synchronization point until the counter becomes zero.

- `latch` helps managing one task with multiple threads

Member Function	Description
`lat.count_down(upd = 1)`	Atomically decrements the counter by `upd` without blocking the caller
`lat.try_wait()`	Returns `true` if `counter == 0`
`lat.wait()`	Returns immediately if `counter == 0`. If not blocks until `counter == 0`
`lat.arrive_and_wait(upd = 1)`	Equivalent to `count_down(upd); wait();`

`bossWorkers.cpp`

Latches and Barriers

- `barrier` helps managing of repeated tasks with multiple threads

Member Function	Description
`bar.arrive(upd)`	Atomically decrements counter by `upd`
`bar.wait()`	Blocks at the synchronization point until the completion step is done
`bar.arrive_and_wait()`	Equivalent to `wait(arrive());`
`bar.arrive_and_drop()`	Decrements the counter for the current and the subsequent phase by one
`std::barrier::max`	Returns the maximum value supported by the implementation

- Callable
  - The constructor gets a callable.
  - In the completion phase, the callabe is executed by an arbitrary thread.

`workers.cpp`