

C++ Club Meeting 102

Gleb Dolgich

2020-03-12

A header-only, tiny and easy to use library for game programming and much more written in modern C++, mainly known for its innovative entity-component-system (ECS) model.

- [GitHub](#) (C++17, MIT)
- [Reddit](#)

- [Article](#)

The C++ Lifetime Profile: How It Plans to Make C++ Code Safer

- Daniel Martin

A new decade, a new tool: libman

- Colby Pike (vector-of-bool)
- Reddit
- GitHub
- Specification

libman is a new level of indirection between package management and build systems.

dds is Drop-Dead Simple build and package manager.

- CppCon 2019: Robert Schumacher “How to Herd 1,000 Libraries”

“Making new friends” idiom by Dan Saks

Wikibooks

The goal is to simplify creation of friend functions for a class template.

```
1 #include <iostream>
2 template<typename T>
3 class Foo {
4     T value;
5 public:
6     Foo(const T& t) { value = t; }
7     friend std::ostream& operator <<(std::ostream& os, const
        Foo<T>& b)
8     {
9         return os << b.value;
10    }
11 };
```

A hidden gem: `inner_product` (1/2)

- Article

A hidden gem: inner_product (2/2)



Conor Hoekstra @code_report

@cjdb_ns & @TartanLlama

4

This makes me so incredibly happy! I literally just yesterday googled, C++17 / C++20 zip to see if they had anything, because I wrote some code in both C++ and [Python](#) and Python was so much more beautiful.

```
int solve(int h, vector<int> w, vector<int> l) {  
    int p = 0;  
    for (int i = 0; i < w.size(); ++i)  
        p = max(p, w[i] - l[i] / 4);  
    return max(0, p - h);  
}  
  
def solve(h, w, l):  
    p = max(a - b/4 for a, b in zip(w, l))  
    return max(0, p - h)
```

29w • 03/12/2018 • 17:47



Conor Hoekstra @code_report

@cjdb_ns & @TartanLlama

Also, I just discovered `std::inner_product` – a beautiful temporary solution to a lack of `zip`.
[#cpp](#) [#inner_product](#)

```
int solve(int h, vector<int> w, vector<int> l) {  
    return max(0, inner_product(begin(w), end(w), begin(l), 0,  
        [](auto a, auto b) { return max(a, b); },  
        [](auto a, auto b) { return a - b / 4; }) - h);  
}
```

27w • 16/12/2018 • 09:30



- Raymond Chen: How can I handle both structured exceptions and C++ exceptions potentially coming from the same source?
 - [Reddit](#)
- Raymond Chen: Can I throw a C++ exception from a structured exception?