Initializing a two dimensional std::vector

So, I have the following:

```
std::vector< std::vector <int> > fog;
```

and I am initializing it very naively like:

```
    for(int i=0; i<A_NUMBER; i++)
    {
            std::vector <int> fogRow;
            for(int j=0; j<OTHER_NUMBER; j++)
            {
                 fogRow.push_back( 0 );
            }
            fog.push_back(fogRow);
    }
```

And it feels very wrong... Is there another way of initializing a vector like this?

c++ vector

edited Jul 15 '13 at 20:28

asked Jul 15 '13 at 20:21

fritzone

16.7k 9 44 100

1 I would recommend a 1d vector implementation for a 2d vector if you don't require more space than `std::vector<int>::max_size()` . Here – andre Jul 15 '13 at 20:34

2 Answers

Use the `std::vector::vector(count, value)` constructor that accepts an initial size and a default value:

```
std::vector<std::vector<int>> fog(
    A_NUMBER,
    std::vector<int>(OTHER_NUMBER)); // Defaults to zero initial value
```

If a value other zero, say `4` for example, was required to be the default then:

```
std::vector<std::vector<int>> fog(
    A_NUMBER,
    std::vector<int>(OTHER_NUMBER, 4));
```

And just to mention uniform initialization introduced in c++11, which permits the initialization of `vector` , and other containers, using `{}` :

```
std::vector<std::vector<int>> fog { { 1, 1, 1 },
                                    { 2, 2, 2 } };
```

edited Jul 15 '13 at 20:39

answered Jul 15 '13 at 20:24

hmjd

92.3k 9 129 193

There is no `append` method in `std::vector`, but if you want to make a vector containing `A_NUMBER` vectors of `int`, each of those containing `other_number` zeros, then you can do this:

```
std::vector<std::vector<int>> fog(A_NUMBER, std::vector<int>(OTHER_NUMBER));
```

answered Jul 15 '13 at 20:25

juanchopanza
169k 17 238 356

about append: just a typo :) thanks! – fritzone Jul 15 '13 at 20:28