| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [Calculators/include](http://docs.google.com/index.html) - LocationComparator.h (source / [functions](http://docs.google.com/LocationComparator.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 18 | 19 | 94.7 % | | Date: | 2012-03-23 |  | Functions: | 0 | 0 | - | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 21 | 32 | 65.6 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #ifndef \_LOCATIONCOMPARATOR\_H  2 : : #define \_LOCATIONCOMPARATOR\_H  3 : :   4 : : #include <algorithm>  5 : : #include <functional>  6 : : #include <stdint.h>  7 : : #include "ComparisonResult.h"  8 : :   9 : : namespace nft{  10 : :   11 : : template<typename \_Tp,  12 : : typename \_FirstCompare,  13 : : typename \_SecondCompare>  14 : : class LocationComparator {  15 : : public:  16 : : inline ComparisonResult  17 : : operator ()(\_Tp inLhsFirst, \_Tp inRhsFirst,  18 : : \_Tp inLhsSecond, \_Tp inRhsSecond);  19 : :   20 : : inline   21 : : LocationComparator(bool inUseFirstCompare)  22 : 6526 : :mUseFirstCompare( inUseFirstCompare ) {  23 : : }  24 : :   25 : : private:  26 : : bool mUseFirstCompare;  27 : :   28 : : };  29 : :   30 : : template<typename \_Tp,  31 : : typename \_FirstCompare,  32 : : typename \_SecondCompare>  33 : : inline ComparisonResult  34 : : LocationComparator<\_Tp, \_FirstCompare, \_SecondCompare>  35 : : ::operator ()(\_Tp inLhsFirst,  36 : : \_Tp inRhsFirst, \_Tp inLhsSecond,  37 : 32630 : \_Tp inRhsSecond) {  38 : :   39 : 6526 : \_FirstCompare fComp;  40 : 6526 : \_SecondCompare sComp;  41 : :   42 : 6526 : if(fComp(inLhsFirst, inRhsFirst))  43 : 2298 : {  44 : : return eCompareTrue;  45 : : }  46 : : else  47 : : {  48 [ + + ][ + + ]: 4228 : if(inLhsFirst == inRhsFirst)  [ + + ][ + + ]  [ + + ][ + + ]  [ + + ][ + + ]  49 : : {  50 : : /\* here TF origins are located within a segment as follows  51 : : \* \_FirstCompare is '<' operator  52 : : \* | segment | y  53 : : \* | | ^  54 : : \* | (TF2) \* | |  55 : : \* | | |--->x   56 : : \* | |   57 : : \* | (TF1) \* | (TF1x < TF2x), (TF1y < TF2y)  58 : : \*/  59 : 1719 : if(true == mUseFirstCompare)  60 : : {  61 : 1719 : if(fComp( inLhsSecond, inRhsSecond))  62 : 134 : {  63 : : return eCompareTrue;  64 : : }  65 : : else  66 : : {  67 [ + + ][ # # ]: 1305 : if( inLhsSecond == inRhsSecond )  [ - + ][ # # ]  68 : 1149 : {  69 : : return eCompareUndecided;  70 : : }  71 : : }  72 : : }  73 : : /\* here TF origins are located within a segment as follows  74 : : \* \_SecondCompare is '>' operator  75 : : \* | segment | y  76 : : \* | | ^  77 : : \* | (TF2) \* | |   78 : : \* | | |--->x  79 : : \* | |   80 : : \* | (TF1) \* | (TF1x > TF2x), (TF1y < TF2y)   81 : : \*/  82 : 2041 : else if(sComp(inLhsSecond, inRhsSecond))  83 : 128 : {  84 : : return eCompareTrue;  85 : : }  86 : : else  87 : : {  88 [ - + ][ # # ]: 152 : if(inLhsSecond == inRhsSecond)  [ - + ][ # # ]  89 : 0 : {  90 : 2817 : return eCompareUndecided;  91 : : }  92 : : }  93 : : }  94 : : }  95 : : return eCompareFalse;  96 : : }  97 : :   98 : : typedef LocationComparator<int,  99 : : std::less<int>, std::greater<int> > NonDescending;  100 : : typedef LocationComparator<int,  101 : : std::greater<int>, std::less<int> > NonAscending;  102 : :   103 : : }  104 : : #endif |

|  |
| --- |
| Generated by: [LCOV version 1.9](http://ltp.sourceforge.net/coverage/lcov.php) |