| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [InputParams/include](http://docs.google.com/index.html) - TableReader.h (source / [functions](http://docs.google.com/TableReader.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 7 | 7 | 100.0 % | | Date: | 2012-03-23 |  | Functions: | 9 | 9 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 0 | 0 | - | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #ifndef NFT\_TABLEREADER\_H  2 : : #define NFT\_TABLEREADER\_H  3 : :   4 : : #include <stdint.h>  5 : : #include "Memory.h"  6 : :   7 : : namespace nft {  8 : :   9 : : enum ArrayMode {  10 : : eColumnMajor, //Matrix is represented using column major form  11 : : eRowMajor //Matrix is represented using row major form  12 : : };  13 : :   14 : : //Class for reading data from table  15 : : template<class T, ArrayMode mode = eColumnMajor>  16 : : class TableReader : public NftNewDeleteBase {  17 : :   18 : : public:  19 : : TableReader(uint32\_t rows, uint32\_t columns, T \* mData);  20 : :   21 : : ~TableReader();  22 : :   23 : : inline uint32\_t  24 : : getRows() const;  25 : :   26 : : inline uint32\_t  27 : : getColumns() const;  28 : :   29 : : //This element returns an element from the table  30 : : //PERF:: Can apply caching scheme if required  31 : : T  32 : : get(const uint32\_t row, const uint32\_t column) const;  33 : :   34 : :   35 : : TableReader(const TableReader<T, mode> & source);  36 : :   37 : : TableReader<T, mode> &  38 : : operator =(const TableReader<T, mode> & source);  39 : :   40 : : protected:  41 : : uint32\_t mRows;  42 : : uint32\_t mColumns;  43 : : T \* mData;  44 : :   45 : : };  46 : :   47 : : template<class T, ArrayMode mode>  48 : : TableReader<T,mode>::TableReader(  49 : : uint32\_t rows, uint32\_t columns, T \* inData)   50 : 7215 : :mRows(rows),  51 : : mColumns(columns),  52 : : mData(inData)  53 : 7215 : {  54 : : }  55 : :   56 : : //Destructor shpuld not call delete on the input  57 : : template<class T, ArrayMode mode>  58 : 7215 : TableReader<T,mode>::~TableReader(){  59 : : }  60 : :   61 : : template<class T, ArrayMode mode>  62 : : inline uint32\_t  63 : : TableReader<T,mode>::getRows() const {  64 : : return mRows;  65 : : }  66 : :   67 : : template<class T, ArrayMode mode>  68 : : inline uint32\_t  69 : : TableReader<T,mode>::getColumns() const {  70 : : return mColumns;  71 : : }  72 : :   73 : : //This element returns an element from the table  74 : : //PERF:: Can apply caching scheme if required  75 : : template<class T, ArrayMode mode>  76 : : T  77 : : TableReader<T,mode>::get(const uint32\_t row,  78 : 99787 : const uint32\_t column) const {  79 : 99787 : if(eColumnMajor == mode)  80 : 16595 : return \*(mData + row + (column\*mRows));  81 : : //Row major  82 : 99787 : return \*(mData + (row\*mColumns) + column);   83 : : }  84 : :   85 : : template<class T,ArrayMode mode>  86 : : TableReader<T,mode>::TableReader(const TableReader<T, mode> & source)   87 : : :mRows(source.mRows),  88 : : mColumns(source.mColumns),  89 : : mData(source.mData)  90 : : {  91 : : }  92 : :   93 : : template<class T, ArrayMode mode>  94 : : TableReader<T, mode> &  95 : : TableReader<T,mode>::operator =(const TableReader<T, mode> & source) {  96 : : if(this != &source)  97 : : {  98 : : mRows = source.mRows;  99 : : mColumns = source.mColumns;  100 : : mData = source.mData;  101 : : }  102 : : return \*this;  103 : : }  104 : :   105 : : typedef TableReader<float> FloatTableReader;  106 : : typedef TableReader<short> ShortTableReader;  107 : : typedef TableReader<double> DoubleTableReader;  108 : : typedef TableReader<float, eRowMajor> RowMajorFloatTableReader;  109 : : typedef TableReader<short, eRowMajor> RowMajorShortTableReader;  110 : : typedef TableReader<double, eRowMajor> RowMajorDoubleTableReader;  111 : :   112 : : } // namespace nft  113 : : #endif |

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