```
//
// Programmer:    Craig Stuart Sapp <craig@ccrma.stanford.edu>
// Creation Date: Fri Jun 16 22:19:18 PDT 2006
// Last Modified: Sat Jun 17 04:29:02 PDT 2006
// Filename:      MzNevermore.h
// URL:           http://sv.mazurka.org.uk/include/MzNevermore.h
// Documentation: http://sv.mazurka.org.uk/MzNevermore
// Syntax:        ANSI99 C++; vamp 0.9 plugin
//
// Description:   DFT spectrogram with independent window and transform size.
//

#ifndef _MZNEVERMORE_H_INCLUDED
#define _MZNEVERMORE_H_INCLUDED

#include "MazurkaPlugin.h"  // Mazurka plugin interface for Sonic Visualiser
#include "MazurkaTransformer.h"
#include "MazurkaWindower.h"


class MzNevermore : public MazurkaPlugin {

   public:

   // plugin interface functions:

                    MzNevermore             (float samplerate);
      virtual      ~MzNevermore             ();

      // required polymorphic functions inherited from PluginBase:
      std::string   getName                 (void) const;
      std::string   getMaker                (void) const;
      std::string   getCopyright            (void) const;
      std::string   getDescription          (void) const;
      int           getPluginVersion        (void) const;

      // optional parameter interface functions
      ParameterList getParameterDescriptors (void) const;

      // required polymorphic functions inherited from Plugin:
      InputDomain   getInputDomain          (void) const;
      OutputList    getOutputDescriptors    (void) const;
      bool          initialise              (size_t channels,
                                             size_t stepsize,
                                             size_t blocksize);
      FeatureSet    process                 (float **inputbufs,
                                             Vamp::RealTime timestamp);
      FeatureSet    getRemainingFeatures    (void);
      void          reset                   (void);

      // optional polymorphic functions from Plugin:
      size_t        getPreferredStepSize    (void) const;
      size_t        getPreferredBlockSize   (void) const;
      size_t        getMinChannelCount      (void) const { return 1; }
      size_t        getMaxChannelCount      (void) const { return 1; }

   // non-interface functions and variables:

   private:

      int    mz_transformsize; // DFT transform size
      int    mz_minbin;        // minimum bin to display
      int    mz_maxbin;        // maximum bin to display
      int    mz_compress;      // for compressing the magnigude range
      int    mz_scale;         // for the vertical scale of freq. axis

      MazurkaTransformer mz_transformer;  // interface FFTW Fourier transforms
      MazurkaWindower    mz_windower;     // interface for windowsing signals

      // input parameters:
      //
      //    "windowsamples"    -- number of samples in audio window
      //    "transformsamples" -- number of samples in transform
      //    "stepsamples"      -- number of samples between analysis windows
      //    "minbin"           -- lowest transform bin to display
      //    "maxbin"           -- highest transform bin to display
      //    "scale"            -- linear or logarithmic scaling of the freqs.

};


#endif // _MZNEVERMORE_H_INCLUDED
```