```
//
// Programmer:    Craig Stuart Sapp <craig@ccrma.stanford.edu>
// Creation Date: Sun Jun 18 00:21:58 PDT 2006
// Last Modified: Sat Jun 24 01:34:13 PDT 2006
// Filename:      MzHarmonicSpectrum.h
// URL:           http://sv.mazurka.org.uk/include/MzHarmonicSpectrum.h
// Documentation: http://sv.mazurka.org.uk/MzHarmonicSpectrum
// Syntax:        ANSI99 C++; vamp 0.9 plugin
//
// Description:   Display a harmonic spectrogram.
//

#ifndef _MZHARMONICSPECTRUM_H_INCLUDED
#define _MZHARMONICSPECTRUM_H_INCLUDED

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


class MzHarmonicSpectrum : public MazurkaPlugin {

   public:

   // plugin interface functions:

                  MzHarmonicSpectrum      (float samplerate);
      virtual     ~MzHarmonicSpectrum      ();

      // 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:

      static void   generateMidiNoteList    (std::vector<std::string>& alist,
                                             int minval = 0,
                                             int maxval = 127);
   private:

      int    mz_harmonics;     // number of harmonics in analysis
      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_method;        // how to calculate the harmonicness of a pitch

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

      // input parameters:
      //
      //    "windowsamples";   -- number of samples in audio window
      //    "stepsamples";     -- number of samples between window starts
      //    "harmonics";       -- number of harmonic to consider
      //    "minpitch";        -- minimum pitch to search
      //    "peakenhance";     -- maximum pitch to search
      //    "method";          -- method for calculating pitch
      //    "compress";        -- dynamic range compression toggle

};


#endif // _MZHARMONICSPECTRUM_H_INCLUDED
```