```
//
// Programmer:    Craig Stuart Sapp <craig@ccrma.stanford.edu>
// Creation Date: Sat Jan 13 05:27:58 PST 2007 (copied over from MzNevermore)
// Last Modified: Sat Jan 13 05:28:13 PST 2007
// Filename:      MzSpectralFlatness.h
// URL:           http://sv.mazurka.org.uk/include/MzSpectralFlatness.h
// Documentation: http://sv.mazurka.org.uk/MzSpectralFlatness
// Syntax:        ANSI99 C++; vamp 0.9 plugin
//
// Description:   Spectral flatness measurement plugin for vamp.
//

#ifndef _MZSPECTRALFLATNESS_H_INCLUDED
#define _MZSPECTRALFLATNESS_H_INCLUDED

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


class MzSpectralFlatness : public MazurkaPlugin {

   public:

   // plugin interface functions:

                    MzSpectralFlatness      (float samplerate);
      virtual      ~MzSpectralFlatness      ();

      // 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                 (AUDIODATA 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 double getArithmeticMean       (std::vector<double>& sequence);
      static double getGeometricMean        (std::vector<double>& sequence);
      static double getSpectralFlatness     (std::vector<double>& sequence);
      static void   smoothSequence          (std::vector<double>& sequence,
                                             double gain);

   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
      double mz_smooth;        // smoothing gain

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

      std::vector<double> flatness_curve;  // store data for smoothing
      std::vector<Vamp::RealTime> flatness_times;  // store data for smoothing

      // 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

};


#endif // _MZSPECTRALFLATNESS_H_INCLUDED
```