

CSL CODE EXAMPLE

// CSL demo to play dynamic band-passed white noise

```
void testDynamicFilters() {  
    float dur = 6.0f;                                // seconds to play  
    WhiteNoise noiz(1.0);                             // noise generator  
    RandEnvelope ctr(3, 1000, 2000, 600);              // filter center/bw random walk envelopes  
    RandEnvelope bw(3, 100, 100, 40);                 // c'tor args: (freq, amp, offset, step)  
    Butter butter(noiz, kBandPass, ctr, bw);          // Butterworth filter (inp, type, freq, bw)  
    center.trigger();                                  // trigger the envelopes  
    bw.trigger();  
    gIO->setRoot(butter);                             // plug the filter into the global IO  
    sleepSec(dur);                                     // sleep for the desired duration  
    gIO->clearRoot();                                  // clear the IO's root to make silence  
}
```



SIREN SMALLTALK TOOLS

- ▶ Since 1984: DoubleTalk, HSTK, MODE and Siren
- ▶ Elements:
 - ▶ the Smoke music representation language (music magnitudes, events, generators, functions and sounds)
 - ▶ voices, schedulers and I/O drivers (real-time and file-based voices, sound, OSC and MIDI I/O)
 - ▶ user interface components for musical applications (UI layout tools, widgets and examples)
 - ▶ several built-in applications (editors and browsers for several kinds of Siren objects)
 - ▶ interfaces to external libraries for audio/MIDI/OSC I/O, and packages such as CSL, SuperCollider and Loris

```

Object
├── AbstractEvent
├── DataSet
├── DurationEvent
├── ActionEvent
├── SoundEvent
├── Function
│   ├── FourierSummation
│   ├── FunctionGraph
│   ├── LinearFunction
│   ├── ExponentialFunction
│   ├── SplineFunction
│   ├── SHARCSample
│   ├── Sound
│   │   ├── ComponentSound
│   │   ├── SampledSound
│   │   ├── FloatSound
│   │   ├── VirtualSound
│   │   ├── CompositeSound
│   │   └── GapSound
│   ├── Spectrum
│   ├── MusicEvent
│   ├── EventList
│   ├── EventGenerator
│   │   ├── Cloud
│   │   │   ├── DynamicCloud
│   │   │   ├── SelectionCloud
│   │   │   │   ├── DynamicSelectionCloud
│   │   │   │   └── ExtDynamicSelectionCloud
│   │   ├── Cluster
│   │   │   ├── Chord
│   │   │   ├── Arpeggio
│   │   │   ├── Roll
│   │   │   ├── Trill
│   │   │   └── Ostinato
│   │   ├── FunctionEvent
│   │   ├── LPCFrame
│   │   └── LPCSound
│   ├── Track
│   ├── Playlist
│   └── SoundFile

```

EVENT GENERATORS AND EVENT MODIFIERS

- ▶ Objects that model “middle-level” musical structures
- ▶ Have flexible c'tors and methods to return their event lists
- ▶ May be used in interactive player threads
- ▶ Examples
 - ▶ Chord, Cluster, Cloud, SelectionCloud, DynamicSelectionCloud
 - ▶ Roll, Ostinato, Trill, Arpeggio, Peal
- ▶ EventLists created by EventGenerators can be processed further with filters, rules, etc.
- ▶ EventModifiers apply functions to event properties

