Real-time audio synthesis with libpd

Peter Brinkmann
Rich Eakin
Modular synthesis

Doepfer A100
Pure Data (Miller Puckette, 1996)
Pd demo
Embedding Pure Data with libpd

- libpd is a thin wrapper that turns Pd into a signal processing library
- Six core developers, many contributors
- Open Source (BSD License)
- Written in C, with bindings for Java, Objective-C, C++, C#, Python, Processing, (Lua)
- No dependencies(!)
- Audio glue for Android, iOS, PortAudio, (Unity3D)
Anatomy of a libpd-based app

**PdBase**
wraps libpd's C API, provides thread synchronization

**PdAudioUnit**
hooks libpd's process function into Core Audio

**PdAudioController**
manages audio session as well as audio unit

---

<table>
<thead>
<tr>
<th>Application code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilities</td>
</tr>
<tr>
<td>Audio glue</td>
</tr>
<tr>
<td>Language bindings</td>
</tr>
<tr>
<td>(may be specific to platform)</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>libpd</th>
</tr>
</thead>
<tbody>
<tr>
<td>(embeddable library wrapper for Pd, written in C)</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Pure Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(written in C)</td>
</tr>
</tbody>
</table>
#include "PdBase.h"

NSString *bundlePath = [[NSBundle mainBundle] bundlePath] ;
void *handle = [PdBase openFile:@"patch.pd" path:bundlePath];

...  
// when done with patch:
[PdBase closeFile:handle];

- The handle is a pointer to an internal data structure representing the patch
- Can open multiple instances of each patch; uniquely identify each with the handle
- A utility class, PdFile, wraps handles in an object-oriented, container-friendly way
PdBase: Sending messages

Or, how to control your Pd patches:

[PdBase sendFloat:440.0 toReceiver:@"freq"];
[PdBase sendBangToReceiver:@"makesound"];

Sending other message types:

+ (int)sendSymbol:(NSString *)symbol toReceiver:(NSString *)receiverName;
+ (int)sendList:(NSArray *)list toReceiver:(NSString *)receiverName;
+ (int)sendMessage...
+ (int)sendNoteOn...
etc.
PdBase: Receiving messages

Get info out of Pd with a class that implements the PdReceiverDelegate protocol:

[PdBase setDelegate:self];
[PdBase subscribe:@"foo"];  // receive messages from '[s foo]'

- (void)receiveFloat:(float)received fromSource:(NSString *)source {
    if ([source isEqualToString:@"foo"]) {
        NSLog(@"foo: %f", received);
    }
}

Related utility class: PdDispatcher
PdAudioController

- Manages audio session and audio unit
- Two kinds of configuration:
  - playback (with or without input)
  - ambient (no input, allows background)
- Both kinds support mixing with other apps
- If you need more fine-grained control of your audio session, you can bypass PdAudioController and use PdAudioUnit directly
**PdAudioController**

**Example audio setup method:**

```objective-c
- (void)setUpPdAudio {
    self.audioController = [[[PdAudioController alloc] init] autorelease];
    PdAudioStatus status = [self.audioController configurePlaybackWithSampleRate:44100
                       numberChannels:2 inputEnabled:NO mixingEnabled:NO];
    if (status == PdAudioError) {
        // Unable to configure audio controller
    } else if (status == PdAudioPropertyChanged) {
        // Partial success; had to change some parameters
    }
}
```

**Start audio:**

```objective-c
- (void)startAudio() {
    self.audioController.active = YES;
}
```
The big picture

- libpd achieves complete separation of concerns between audio and app development
- Pd acts as a scripting language for audio
- libpd erases the distinction between prototype and production code
- libpd lets you deploy synthesis code as easily as media files
- libpd lowers the barrier to entry for generative audio
Showcase

- **Inception The App**
  - Experience your life as a dream.
  - The app that changes your perception of reality.
  - Available on the App Store.

- **The Dark Knight Rises Z+**
  - Augmented soundtrack experience.
  - Available on the App Store.

- **Pugs Luv Beats**
  - Illustrated character with mountains and hills.

- **Bad Hotel**
  - LUCKY FRAME
  - Illustrated building.

- **NodeBeat**
  - Illustrated composition with circular shapes and dots.
Tutorial: Touch-Based Synthesis

● **App 1: Basic Monophonic Synth**
  ○ patch: additive synthesis with wavetable lookup
  ○ receives a float from pd for cpu load

● **App 2: SlidePad Polyphonic Synth**
  ○ every touch is a unique 'voice' (patch)
  ○ switch between additive or subtractive synthesis
  ○ controls:
    ■ Pitch Quantize
    ■ Octave Shift
    ■ Low-pass filter
... to Xcode!
To Take Home

● When using many resources (i.e., polyphony), allocate up front
● Store touches in NSDictionary with value = [NSValue valueWithPointer:touch]
● Prepare to spend a lot of time fine tuning audio levels

SlidePad and other sample apps are available at github.com/libpd/pd-for-ios.
Resources

libpd:
libpd.cc
github.com/libpd
Brinkmann, 2012, "Making Musical Apps"

Sound Synthesis with Pd:
Farnell, 2010, "Designing Sound"

Forums:
http://createdigitalnoise.com/categories/pd-everywhere
http://lists.puredata.info/listinfo/pd-list
Acknowledgments

- Miller Puckette
- RjDj team, especially Martin Roth
- the many libpd contributors

Thank you!